

abc

N° 18

COURS
D'INFORMATIQUE
PRATIQUE
ET FAMILIALE

INFORMATIQUE



Les disques Winchester

Questions de langages

L'Apple Ile décortiqué

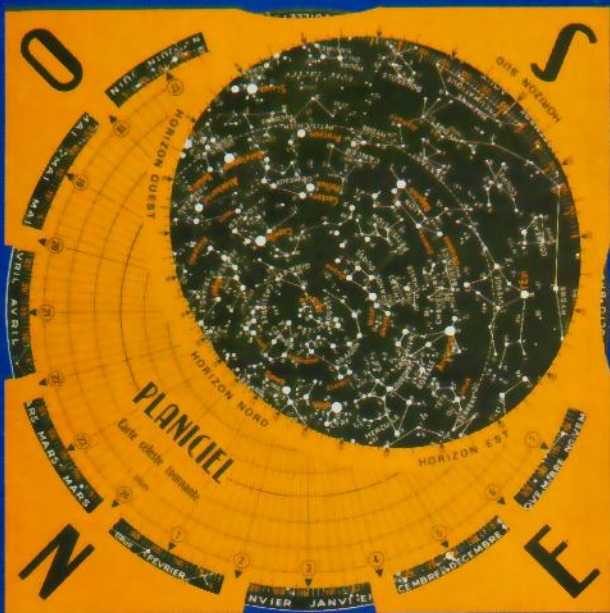
Transport et ordinateur

EDITIONS
ATLAS

M6062-18-12F

85FB-3,80FS- $\$$ 1.95

INITIATION A L'ASTRONOMIE



dimensions réelles : 30 x 30 cm

* **CARTE TOURNANTE**
- 840 étoiles
- 60 amas ou nébuleuses

* **ASTRONOMIE ÉLÉMENTAIRE**
- Observations contrôlées
- Premières notions

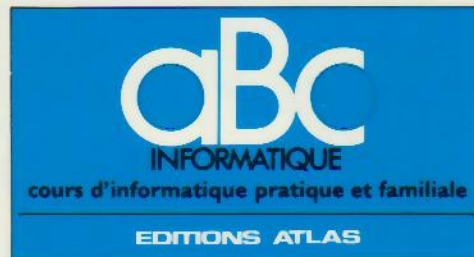
* **COORDONNÉES ÉQUATORIALES**

- Angles horaires
- Temps sidéral
- Écliptique
- Zodiaque

* **CARTE LUNAIRE**

- Cent détails du relief
- identifiables à l'œil nu ou
- aux jumelles.

Documentation : **PLANICIEL - 625 boulevard des Lucioles
83700 SAINT RAPHAËL - ANTHÉOR.
FRANCE - Tél. (94) 44.80.66**



Édité par ÉDITIONS ATLAS s.a., tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15. Tél. : 538-52-70.

Belgique : ÉDITIONS ATLEN s.a., Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, Montréal Nord.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, Mezzovico.

Realisé par EDENA s.a., 29, boulevard Edgar-Quinet, 75014 Paris. Tél. : 320-15-01.

Direction éditoriale : J.-Fr. Gautier. Service technique et artistique : F. Givone et J.-Cl. Bernar. Iconographie : J. Pierre. Correction : B. Noël.

Publicité : Anne Cayla. Tél. : 202-09-80.

VENTE

Les numéros parus peuvent être obtenus chez les marchands de journaux ou, à défaut, chez les éditeurs, au prix en vigueur au moment de la commande. Ils resteront en principe disponibles pendant six mois après la parution du dernier fascicule de la série. (Pour toute commande par lettre, joindre à votre courrier le règlement, majoré de 10 % de frais de port.)

Pour la France, s'adresser à ÉDITIONS ATLAS, tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15. Tél. : 538-52-70.

Pour les autres pays, s'adresser aux éditeurs indiqués ci-dessous.

SOUSCRIPTION

Les lecteurs désirant souscrire à l'ensemble de cet ouvrage peuvent s'adresser à :

France : DIFFUSION ATLAS, 3, rue de la Tave, 28110 Lucé. Tél. : (37) 35-40-23.

Belgique : ÉDITIONS ATLEN s.a., 55, avenue Huart-Hamoir, 1030 Bruxelles. Tél. : (02) 242 39 00. Banque Bruxelles-Lambert, compte n° 310-0018465-24 Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, 11450 boulevard Albert-Hudon, Montréal Nord, H 1G 3J9.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, zona industriale, 6849 Mezzovico-Lugano. Tél. : (091) 95-27-44.

RELIEZ VOS FASCICULES

Des reliures mobiles, permettant de relier 12 fascicules, seront en vente en permanence chez votre marchand de journaux.

ATTENTION : ces reliures, présentées sans numérotation, sont valables indifféremment pour tous les volumes de votre collection. Vous les numéroterez vous-même à l'aide du décalque qui est fourni (avec les instructions nécessaires) dans chaque reliure.

En vente tous les vendredis. Volume II, n° 18.

ABC INFORMATIQUE est réalisé avec la collaboration de Trystan Mordrel (secrétariat de rédaction), J.-P. Bourcier (coordination), St.-André Laroche (traduction), Ghislaine Goullier (fabrication), Marie-Claire Jacquet (iconographie), Patrick Boman (correction).
Crédit photographique, couverture : Photo CRDP.

Directeur de la publication : Paul Bernabeu. Imprimé en Italie par I.G.D.A., Officine Grafiche, Novara. Distribution en France : N.M.P.P. Tax. Dépôt légal : mai 1984, 4845. Dépôt légal en Belgique : D/84/2783/27.

© Orbis Publishing Ltd., London.
© Éditions Atlas, Paris, 1984.

A NOS LECTEURS

En achetant chaque semaine votre fascicule chez le même marchand de journaux, vous serez certain d'être immédiatement servi en nous facilitant la précision de la distribution. Nous vous en remercions d'avance.

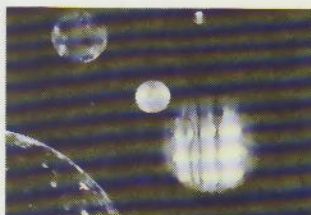
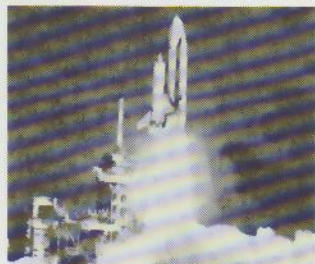
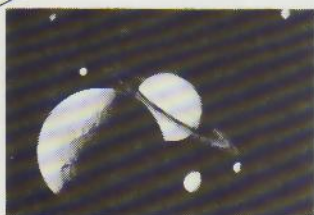
Les Éditions Atlas

LES DIAPOSITIVES DE LA NASA

Pour projecteur standard

10 séries disponibles avec commentaires de vues

OFFRE ILLIMITÉE



D1/APOLLO 11. Les premiers hommes sur la lune, juillet 1969, série de 22 diapositives couleur : 109 F (port inclus).

D2/APOLLO 17. Dernière mission lunaire, décembre 1972, utilisation de la jeep lunaire, série de 35 diapositives couleur : 139 F (port inclus).

* Comportant la photo de la terre la plus claire, jamais prise.

D3/JUPITER VOYAGER 1 ET 2. La planète géante de notre système solaire. Série de 31 diapositives couleur : 139 F (port inclus).

D4/SATURNE VOYAGER 1. La planète la plus spectaculaire de notre système solaire, série de 18 diapositives couleur : 99 F (port inclus).

D5/COLUMBIA. Le premier vol orbital du « camion de l'espace », série de 27 diapositives couleur : 119 F (port inclus).

Le lot D1 à D5 : 590 F

LE LOT DE 10 SÉRIES : 1060 F seulement (port inclus).

D6/VIKING 1 ET 2 SUR MARS. La mission américaine sur la planète rouge en 1976, série de 20 diapositives couleur : 99 F (port inclus).

D7/LE SOLEIL. Les photos étonnantes de notre soleil prises par Skylab et les plus grands observatoires du monde, série de 20 diapositives couleur et noir et blanc : 99 F (port inclus).

D8/LES GALAXIES. Un éventail coloré des différents types de galaxies peuplent notre univers, série de 20 diapositives couleur et noir et blanc : 99 F (port inclus).

D9/LES CONSTELLATIONS DU CIEL. comprenant les 12 constellations zodiacales, série de 20 diapositives : 99 F (port inclus).

D10/LES COLONIES SPATIALES DU FUTUR.

Série fiction décrivant les grands projets de colonisation humaine de l'espace, série de 20 diapositives couleur : 99 F (port inclus).

Le lot D6 à D10 : 490 F

Delai de livraison 20 jours minimum. Les chèques ne sont pas débités avant envoi.

BON DE COMMANDE

à retourner accompagné de votre règlement à :
GALAXY CONTACT BP 26 62101 CALAIS Cedex FRANCE

Nom Prénom

Adresse

Réf. de la commande

Je désire recevoir votre catalogue complet (cartes postales, diapositives, posters, choix de plus de 65 photos N.A.S.A. couleur), je joins 4 timbres à mon envoi.



Chemin faisant



Transport public

Le VAL, premier métro entièrement automatique, traverse la ville de Lille, dans le nord de la France, d'est en ouest. L'automatisation de ce système de transport sans chauffeur permet un passage d'une rame toutes les minutes aux heures de pointe. C'est une première mondiale. Sa vitesse commerciale est très élevée (35 km/h arrêts compris). Tout le système est contrôlé à partir des postes de commande situés en début de ligne. (Cl. Matra.)

Le transport des biens et des personnes d'un endroit à un autre n'est pas une affaire simple dans notre monde surchargé. L'ordinateur est une aide précieuse.

Pour aller d'Europe en Australie, il fallait compter plus de trois mois au début du XIX^e siècle. Aujourd'hui, le même voyage ne prend pas plus d'une demi-journée. Ce miracle n'a pu se réaliser sans l'arrivée de nouvelles technologies utilisant l'informatique comme outil de régulation et de contrôle.

L'avion est le moyen de transport qui pose le plus de problèmes. L'aéroport Charles-de-Gaulle à Paris, par exemple, est le théâtre de plusieurs centaines de vols chaque jour, avec près de cent mouvements d'avion en une heure aux moments de pointe. Sans le concours d'un ordinateur pour contrôler cette activité, un tel système ne pourrait fonctionner.

Prenons le cas d'un voyageur qui désire se rendre de Paris à New York. Entre l'agence de voyages où le billet est acheté et la place réservée (en utilisant le système Télétel pour la réservation directe auprès de la compagnie aérienne) et l'arrivée à l'aéroport Kennedy, plus de quinze ordinateurs différents auront éventuellement été sollicités pour ce voyage. Regardons de plus près comment ils interviennent.

Une première étape apparaît au niveau de l'appareil aéronautique lui-même. Les avions civils modernes coûtent extrêmement chers. Pour rentabiliser au plus vite les investissements qu'elles doivent consentir, les compagnies doivent « garder » leurs avions dans les meilleures conditions possibles. La clé du problème réside

dans l'organisation de la maintenance. Après un nombre prédéterminé d'heures de vol, l'avion doit retourner à sa base pour faire une « toilette ». Les techniciens ont alors accès à toute son histoire, enregistrée en mémoire d'ordinateur, depuis le premier jour de sa construction jusqu'à l'ensemble de ses contrôles successifs. Les enregistrements donnent les détails des différentes opérations que l'appareil a subies, toutes ses performances, ses consommations de carburant, enfin toutes les informations susceptibles d'apporter des éléments intéressants pour son avenir. L'utilisateur d'ordinateur domestique peut entreprendre une démarche similaire — avec peut-être moins de détails — pour assurer lui-même la maintenance de son automobile.

Ce n'est que lorsque le tableau des opérations de maintenance est totalement mis à jour que l'avion peut retourner en service. Il retrouve alors sa place dans un autre système informatique, celui qui définit les plans de vol, organise les divers points de ravitaillement aux étapes, arrange les menus à bord, et éventuellement les films et la musique, bref celui qui prend en charge tous les éléments nécessaires pour assurer le transport de trois cents ou quatre cents personnes autour d'une moitié du globe.

Un autre système de contrôle informatisé opère à l'intérieur de l'aéroport lui-même. Les responsables de ces organismes doivent en effet



faire face à d'importantes demandes pour obtenir des autorisations d'envol ou d'atterrissage. Quelques minutes de retard signifient beaucoup d'argent perdu. Seule une gestion de ces opérations par ordinateur permet de limiter les frais. Le même système informatique de l'aéroport se charge également d'appeler les passagers, d'afficher les départs et les arrivées.

Mais, avant que les passagers arrivent à l'aéroport, le pilote aura déjà établi son plan de vol et l'aura transmis aux contrôleurs aériens. Curieusement, les services de contrôle aérien n'utilisent que partiellement les ordinateurs. Grâce au système radar, les contrôleurs suivent l'avion et communiquent verbalement avec le pilote. Ils voient le signal de l'avion sur leurs écrans, et l'altitude et la direction sont directement transmises par l'intermédiaire de l'ordinateur de bord. Un degré supplémentaire d'intervention de l'ordinateur dans la tâche du contrôleur apparaît sous forme de papier imprimé sur lequel est inscrite une partie du plan de vol, similaire à celui dont dispose le pilote. Les informations contenues sur ce document permettent au contrôleur de guider l'avion à l'intérieur de sa zone pour trouver le chemin le plus rapide et le plus économique.

Le rail est un autre domaine des transports où l'ordinateur est fortement utilisé. Si le travail des aiguilleurs de chemins de fer ne semble pas aussi compliqué que celui des contrôleurs aériens, le rapprochement est pourtant évident. Dans les deux cas il s'agit de suivre et guider à

l'intérieur d'une zone le trafic des passagers et des marchandises, en toute sécurité et au coût le plus bas. Depuis de nombreuses années, la S.N.C.F. fait appel à l'informatique pour gérer son immense réseau. L'ordinateur peut suivre le déplacement d'un wagon de marchandises, par exemple, à partir de son numéro d'identification. Si ce wagon est demandé à un endroit particulier, l'ordinateur l'affecte à un train correspondant à la destination. Une fois arrivé, sa nouvelle situation est enregistrée pour être ultérieurement réexploitée. Si l'on tient compte de l'énorme parc roulant dont dispose la S.N.C.F., l'usage de l'ordinateur n'est pas superflu.

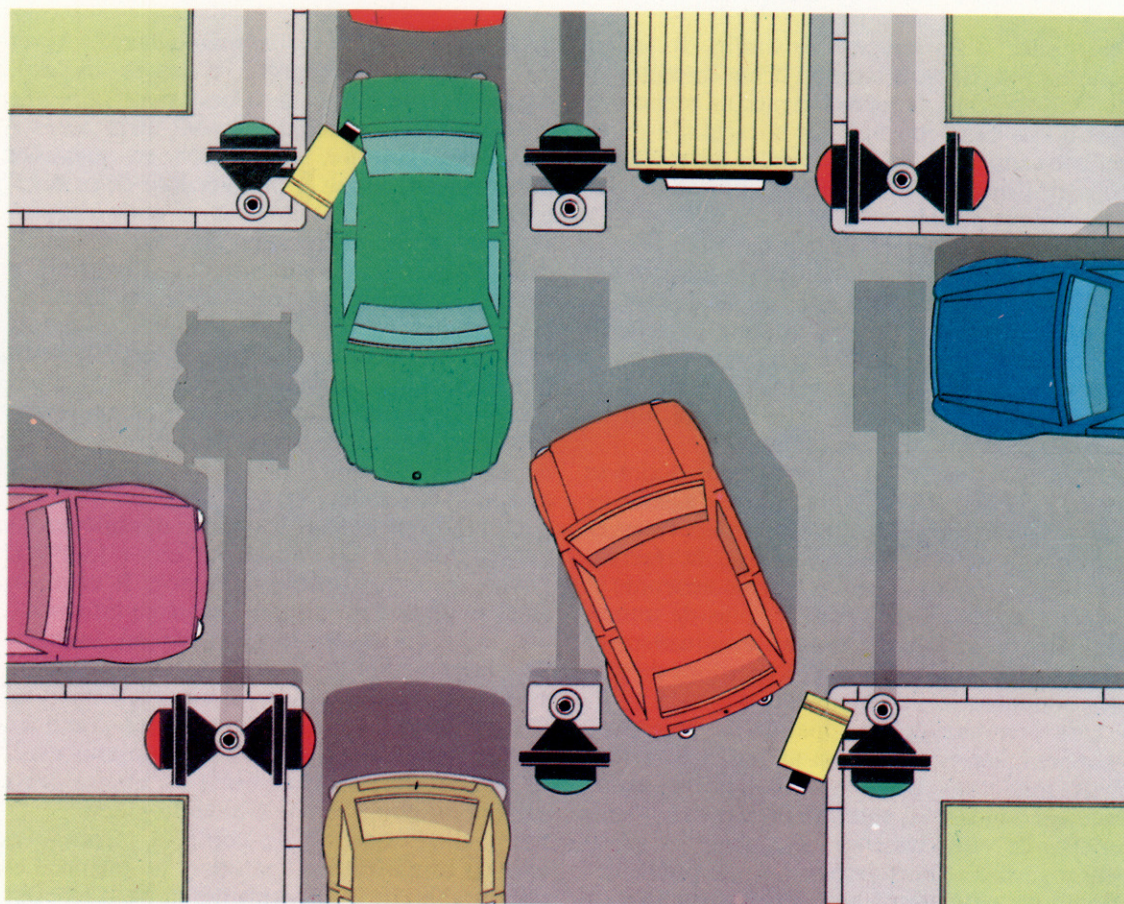
Beaucoup de problèmes sont communs aux opérations à l'intérieur d'un aéroport et à celles qui apparaissent entre les stations de chemin de fer. Mais, dans ces derniers cas, l'ordinateur semble aller plus loin. De nombreuses expériences se déroulent actuellement et font appel aux micro-ordinateurs pour donner des informations aux passagers, par exemple dans les stations sans personnel (dans certains cas, l'arrivée et le départ des trains sont annoncés par voix synthétique).

Mais l'ordinateur est également à l'origine des trains sans conducteur. Une première mondiale eut d'ailleurs lieu à Lille il y a quelques années : le VAL fut le premier métro entièrement automatique à circuler commercialement. La seule ligne actuellement opérationnelle (d'autres sont en préparation) se développe sur 16,5 km et comporte dix-huit stations. Le

Feu vert

Depuis longtemps, la régulation des feux de croisement est employée par les spécialistes pour faciliter l'écoulement du trafic routier, en particulier dans les rues des grandes villes. Aujourd'hui, les indicateurs lumineux peuvent suivre individuellement la densité du trafic dans leur voisinage en se fondant sur les détections des radars à effet doppler enregistrées en mémoire d'ordinateur. Les changements de lumière peuvent donc être ajustés aux conditions du moment.

(Cl. Roy Ingram.)





Attention au départ..

L'aspect le plus spectaculaire du système informatique dans un aéroport ou une gare tient, sans aucun doute, dans le tableau d'affichage des arrivées et des départs. Les panneaux sont constamment mis à jour par l'ordinateur central. (Cl. Air France.)

système est contrôlé à partir d'un poste de commande situé en début de ligne, ce qui permet de gérer tout ce qui se passe dans le métro. Une transmission vidéo complète le système (huit mille télémessures sont réalisées toutes les deux secondes). Ce métro sans conducteur est donc télésurveillé grâce à une électronique embarquée permettant un dialogue entre la voie et les antennes de contrôle.

Faire marcher un train sans conducteur en toute sécurité est relativement facile dans la mesure où il roule sur des rails bien rigides. Il en va tout autrement pour les véhicules routiers. Les ordinateurs sont néanmoins utilisés dans les transports publics sur route et dans les opérations de fret. Ils permettent d'établir les horaires et les routes des véhicules appartenant à un réseau urbain. C'est une tâche très compliquée dans une ville comme Paris où il s'agit d'intégrer dans un même service public les bus, le R.E.R. et le métro.

Le problème est de garder assez de véhicules en circulation pour que les passagers n'attendent pas trop longtemps, sans que pour autant un nombre trop grand de véhicules viennent à créer des dépenses inutiles. Ce genre de casse-tête relève des statistiques, ce que précisément les ordinateurs aiment à résoudre !

Un autre problème du même genre est d'établir, à partir de méthodes statistiques, le meilleur itinéraire pour les véhicules de livraison, ou — variante — comment assurer la répartition la mieux adaptée de divers véhicules comme les taxis ou les cars de police.

Une des expériences particulièrement intéressantes d'utilisation de l'ordinateur dans le transport public se trouve dans la banlieue de Hanovre (R.F.A.). Un système électronique permet d'appeler des minibus à partir de « mini-

postes » d'interrogation pour qu'ils viennent vous prendre ; par la même occasion, vous donnez votre destination. De votre poste d'appel, vous recevez un papier sur lequel sont inscrits l'heure d'arrivée (jamais plus de cinq minutes d'attente), la durée du voyage (il faut tenir compte des autres passagers) et le prix.

Plus de 20 % du commerce mondial correspond au transport de passagers et de marchandises. L'usage de l'ordinateur est dans ce domaine souvent plus en avance que dans d'autres. Sans aucun doute, il a contribué à la croissance importante des échanges internationaux. Certes, la plupart des exemples précédents mettent en œuvre des ordinateurs puissants, mais, dans certains cas, des ordinateurs familiaux s'en tireraient très bien. Des logiciels voient le jour pour mettre en place des programmes d'horaire, de distribution, etc.

Navires marchands

L'industrie navale utilise les ordinateurs dans des domaines très variés. Mais une des applications est particulièrement importante : il s'agit des services de conteneurisation. Les micro-ordinateurs jouent un rôle important dans l'arrangement et l'organisation des conteneurs au poste terminal, ainsi que dans leur chargement sur le bateau afin de répartir correctement les poids. (Cl. Motor Ship.)



Divers langages

Le BASIC, d'une construction assez logique et familière, est donc facile à apprendre, mais il ne fait pas le poids comparé à d'autres langages.

A moins que ce ne soit un Jupiter Ace, il est presque certain que votre ordinateur domestique dispose du BASIC comme langage intégré. Mais cela ne veut pas dire que vous êtes limité à ce choix. Si le BASIC est un langage particulièrement facile à apprendre, il en existe d'autres, beaucoup plus efficaces pour écrire des applications spécifiques. Pour installer ces langages sur votre ordinateur, vous devrez soit remplacer les ROM qui renferment l'interpréteur BASIC, soit charger le nouveau langage en RAM — dans ce cas, vous avez besoin d'une machine disposant d'une mémoire suffisante pour loger vos programmes. Quelques ordinateurs domestiques,

BASIC-FRANÇAIS

FRANÇAIS-BASIC

Le BASIC fut développé à partir du FORTRAN (un des premiers langages de programmation évolués et toujours le langage le plus populaire pour les applications scientifiques et techniques) à titre d'introduction à la programmation pour les étudiants à l'université. Parce qu'il était destiné à permettre une autoformation, le BASIC est généralement interprété et non compilé. Cette caractéristique a fait de lui le langage intégré de la plupart des ordinateurs domestiques. La mise en place de langages interprétés est peu coûteuse.

Le BASIC est maintenant devenu un langage puissant, mais son efficacité est limitée par son grand nombre de versions non standard (il existe un BASIC propre à chaque machine) et par son manque de structures spécialisées de données et de commandes. Ces restrictions signifient que les programmeurs autodidactes peuvent développer une mauvaise technique de programmation, et qu'une bonne technique de programmation peut être difficile à utiliser pour des problèmes spécifiques en BASIC. Ce court programme illustre la simplicité mais aussi les restrictions du BASIC :

```
100 INPUT « Quel est votre nom ? »; N$
200 INPUT « Et quel âge avez-vous ? »; A
300 FOR K = 1 TO A
400 PRINT K, « Bonjour »; N$
500 NEXT K
600 INPUT « Une autre fois ? »; O$
700 IF LEFT$(O$,1) = O$ THEN GOTO 100
800 PRINT « Au revoir »; N$
900 END
```

PASCAL-FRANÇAIS

FRANÇAIS-PASCAL

Le PASCAL fut développé au début des années soixante-dix pour remplacer le BASIC. Ses structures de données et de commandes sont inspirées du groupe de langages FORTRAN-ALGOL et ont pour objet d'encourager l'étudiant à approcher la programmation de façon systématique, afin d'écrire des programmes bien structurés et faciles à comprendre. C'est la seule façon de développer une bonne technique de programmation. Mais cette approche implique plus de travail pour le débutant en raison de la discipline imposée par ce langage et parce qu'il est compilé et non interprété. Néanmoins, les programmes en PASCAL sont généralement assez élégants; ils sont développés assez rapidement et apparaissent beaucoup plus faciles à comprendre.

Voici l'équivalent PASCAL du programme BASIC :

```
VAR
NOM      : PACKED
          ARRAY (1..30)
          OF CHAR;
ÂGE,COMPTE : INTEGER
RÉPONSE   : PACKED
          ARRAY (1..3)
          OF CHAR;
EXECUT    : BOOLEAN;

BEGIN
EXECUT := TRUE;
WHILE EXECUT DO
    BEGIN
    WRITE (« Quel est votre nom ? »);
    READLN(NOM);
    WRITE (« Et quel âge avez-vous ? »);
    READLN(ÂGE);
    FOR COMPTE := 1 TO ÂGE DO
        WRITE(COMPTE : 3, « Bonjour »;
        10,NOM);
    WRITE (« Une autre fois ? »);
    READLN(RÉPONSE);
    IF RÉPONSE(1) = « N »
    THEN EXECUT := FALSE;
    END
    WRITELN (« Au revoir », NOM);
    END
END
```

COMAL-FRANÇAIS

FRANÇAIS-COMAL

Le COMAL fut développé pour combiner la facilité du BASIC avec les puissantes structures du PASCAL et la discipline requise pour son approche. Il ressemble donc à ces deux langages. Il a peut-être servi de modèle pour développer le BASIC du BBC, qui est presque un nouveau langage par lui-même. Le COMAL est très populaire dans l'enseignement en Scandinavie (où il a été créé), mais il semble peu probable qu'il réussisse à remplacer ses prédécesseurs comme langage d'introduction à la programmation.

Voici le programme « Bonjour » écrit en COMAL :

```
100 EXECUT := TRUE
200 WHILE EXECUT DO
300 INPUT « Quel est votre nom ? »; N$
400 INPUT « Et quel âge avez-vous ? »; A
500 REPEAT
600   FOR K = 1 TO A DO
700     PRINT K, « Bonjour »; N$
800   NEXT K
900 INPUT « Une autre fois ? »; O$
1000 IF O$ = « N » THEN EXECUT := FALSE
1100 ENDWHILE
1200 PRINT « Au revoir »; N$
```




LISP-FRANÇAIS

FRANÇAIS-LISP

Le LISP fut développé au début des années soixante comme un langage de traitement de liste. Depuis, il a été largement utilisé dans le domaine de l'intelligence artificielle qui implique des recherches et des comparaisons continues dans les listes de données, des relations et des réponses. Contrairement au BASIC, où les instructions et les procédures sont exécutées de façon séquentielle, le LISP est un langage « fonctionnel », avec lequel le jeu de commandes élémentaires peut être utilisé pour composer des fonctions plus sophistiquées dont les noms sont définis par le programmeur. Par exemple :

(SETQ TABLEAU1 '(4 7 2 5 1))

crée une liste nommée TABLEAU 1 dont les éléments sont les chiffres 4, 7, 2, 5 et 1.

(CAR TABLEAU1)

donne le premier élément de TABLEAU 1 (ici 4).

(CDR TABLEAU1)

donne la liste de TABLEAU1 où le premier élément a été retiré — (7 2 5 1) dans ce cas.

SETQ TABLEAU1(CDR TABLEAU1)

transformera TABLEAU1

en une copie de lui-même excluant son premier élément. Le LISP se prête également à des applications « récursives » — des problèmes qui à la suite d'une simple fonction sont réduits à un problème plus petit mais identique.

FORTH-FRANÇAIS

FRANÇAIS-FORTH

Le FORTH étant un langage fonctionnel interactif, il ressemble au LOGO, mais avec une distinction importante : il est le premier langage autre que le BASIC à avoir été choisi comme langage intégré dans un ordinateur domestique — le Jupiter Ace. Ce langage consiste en un nombre de fonctions définies, dites « primitives » ; il offre la possibilité de définir de nouvelles fonctions à partir de ces fonctions « primitives ». Les opérations mathématiques sont effectuées à l'aide de piles en FORTH, ce qui signifie que la mémoire de l'ordinateur est traitée comme une liste de données croissante ou décroissante. Résultat : la dernière opération est toujours à la tête de la liste. Une autre conséquence de cette approche est que la notation algébrique n'est pas utilisée. Au lieu d'écrire $(12 + 4)/2$ pour trouver la moyenne de 12 et 4, en FORTH vous devez écrire $12\ 4\ +\ 2/$, qui représente la même opération en notation polonaise inversée.

Tout cela fait du FORTH un langage très différent qui nous oblige à adopter une nouvelle approche pour résoudre les problèmes et pour planifier les traitements informatiques. C'est presque un pas en arrière dans la hiérarchie des langages évolués. Ce fragment de programme FORTH définit deux nouveaux mots, CRI et CHŒUR :

:CRI (imprime « CÉSAR! »)

« CÉSAR! »;

:CHŒUR utilise CRI dans une boucle

0 DO CRI LOOP;

Maintenant, taper n CHŒUR entraînera l'affichage de CÉSAR! n fois à l'écran.

LOGO-FRANÇAIS

FRANÇAIS-LOGO

Le LOGO fut développé par un psychologue travaillant à un projet d'intelligence artificielle. Ce langage ressemble au FORTH au niveau de l'interaction et de l'utilisation de fonctions primitives qui peuvent composer des fonctions définies par l'utilisateur. Mais son principe fondamental est le suivant : la meilleure façon d'apprendre à faire quelque chose est de l'enseigner à quelqu'un d'autre — à l'ordinateur. On considère le LOGO comme un langage révolutionnaire qui laisse entrevoir une toute nouvelle façon d'apprendre à penser aux enfants.

Le LOGO est aussi appelé langage de la « tortue » car il permet souvent de commander un petit robot sur roue appelé tortue.

Voici une portion de programme LOGO qui dessine une maison symbolique à l'aide d'un carré et d'un triangle de dimensions spécifiées.

TO LONGUEUR TRIANGLE
RÉPÈTE 3 (AVANCE : LONGUEUR A DROITE 120)

END

TO LONGUEUR CARRÉ
RÉPÈTE 4 (AVANCE : LONGUEUR A DROITE 90)

END

TO LONGUEUR MAISON
DROITE 30

TRIANGLE : LONGUEUR

GAUCHE 90

CARRÉ : LONGUEUR

END

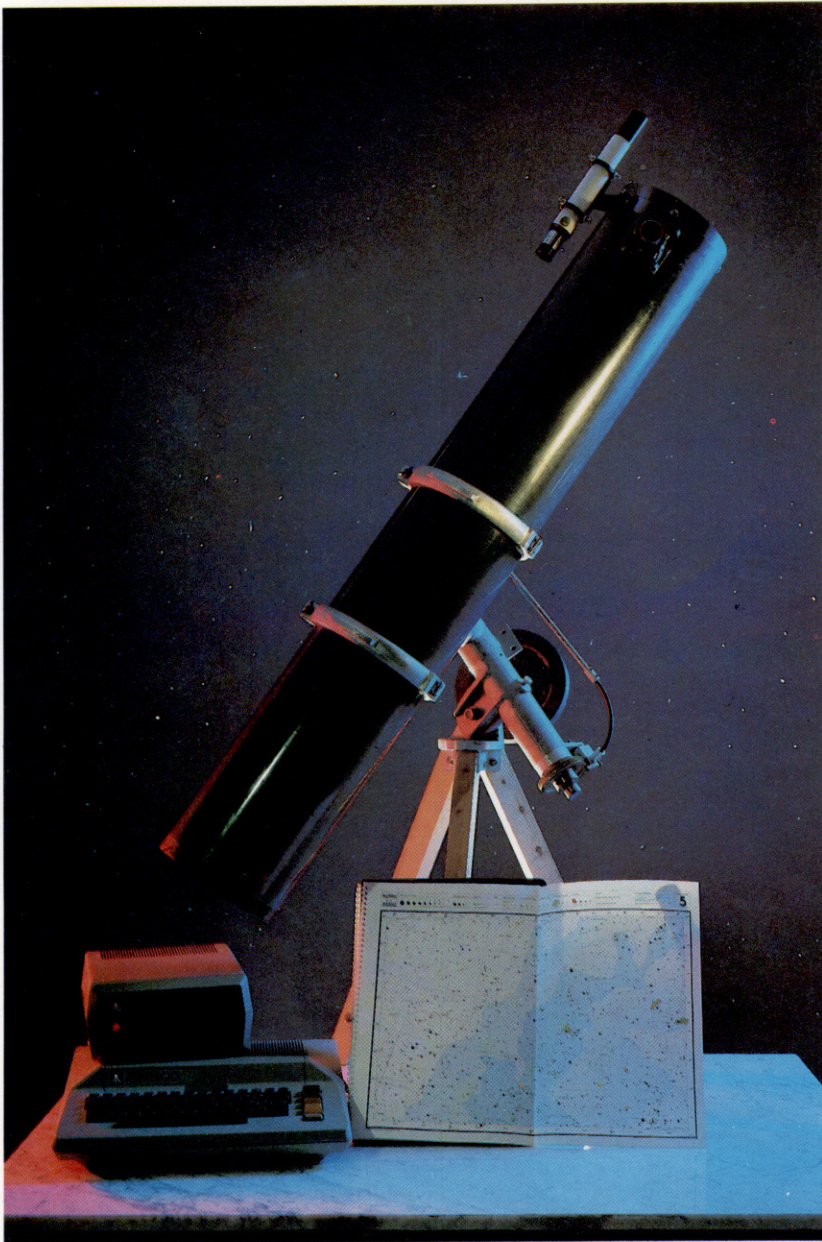
Maintenant, taper MAISON 15 dessinera une maison avec un côté d'une longueur de 15 unités.

comme le Sharp MZ-711, ont prévu ce problème en permettant également de charger l'interpréteur BASIC à partir d'une cassette.

Dans ces pages, nous vous présentons une vue d'ensemble des langages de programmation les plus répandus qui sont disponibles sur les ordinateurs domestiques. Tout comme pour les langages humains, plus vous connaîtrez de langages de programmation, plus il sera facile d'en adopter un nouveau.

Plans sur la comète

Les ordinateurs ont deux utilisations principales dans le domaine de l'astronomie : constituer une base de données des corps observés et calculer leur position afin de faciliter l'orientation des télescopes.



Lumière de guidage

Un ordinateur domestique peut facilement gérer une base de données des emplacements d'étoiles, et peut effectuer rapidement les calculs nécessaires. Avec l'addition de moteurs pas à pas, il peut même positionner le télescope. (Cl. Marcus Wilson-Smith.)

Quand vous regardez le ciel la nuit et que vous fixez Proxima Centauri, l'étoile la plus proche, vous la voyez, en fait, telle qu'elle apparaissait il y a quatre ans, puisque c'est le temps nécessaire pour que sa lumière nous parvienne. Pendant ces quatre années, la micro-informatique a énormément évolué et s'est répandue dans les foyers. L'astronomie est un domaine qui utilise entièrement le potentiel de gestion de données, de calcul et de robotique d'un ordinateur domestique.

Le travail de l'astronome a trois aspects : l'observation initiale du corps céleste, la manipulation des données obtenues lors de cette observation et l'analyse significative de ces données. L'ordinateur peut être utile dans chacune de ces tâches.

Examinons d'abord comment un ordinateur domestique peut faciliter l'exécution des calculs nécessaires pour construire l'outil de base qu'est le télescope. La conception et la disposition des lentilles et des miroirs d'un télescope sont déterminantes pour la qualité de l'image finale, et peuvent être calculées mathématiquement afin d'obtenir les meilleurs résultats possibles. Les astronomes amateurs aiment souvent construire leurs propres systèmes optiques. Avant la large diffusion des ordinateurs domestiques, il était souvent plus rapide et plus simple de réaliser un assemblage expérimental que d'entreprendre des calculs complexes pour la mise au point des schémas optiques. Avec un ordinateur, les calculs qui prenaient une semaine sont exécutés en quelques minutes.

La localisation d'une étoile dans le ciel est un problème qu'un ordinateur aide également à résoudre. Les étoiles ne sont pas des objets fixes. Au cours de la nuit, elles se déplacent dans le ciel (en raison de la rotation de la Terre), et leurs positions varient selon les saisons. La méthode de localisation d'une étoile est celle des longitudes et des latitudes, utilisée en géographie. Si nous imaginons un système de coordonnées projeté sur la surface interne du ciel nocturne, chaque objet céleste peut être localisé au moyen de deux coordonnées nommées *déclinaison* et *ascension droite*.

Chaque objet peut être marqué sur une carte céleste selon ses coordonnées, et la combinaison des cartes individuelles forme un atlas du ciel. De tels atlas sont des outils très importants pour observer des planètes et autres objets qui se déplacent contre la toile de fond des étoiles « fixes », ou pour découvrir de tout nouveaux objets, comme des comètes. Ces atlas célestes ont été traduits en bases de données destinées aux ordinateurs domestiques. Ces bases incluent également des informations relatives à la brillance, ou luminosité, des corps célestes individuels, à la nature spectrale de la lumière qu'ils émettent (obtenue lorsque la lumière traverse un prisme ou par analyse au spectromètre), ainsi qu'au type et à l'âge de l'étoile. Toutes ces informations peuvent s'afficher sur l'écran de l'ordinateur, sous la forme de cartes illustrant le ciel sous n'importe quelle latitude et à tout moment désiré.

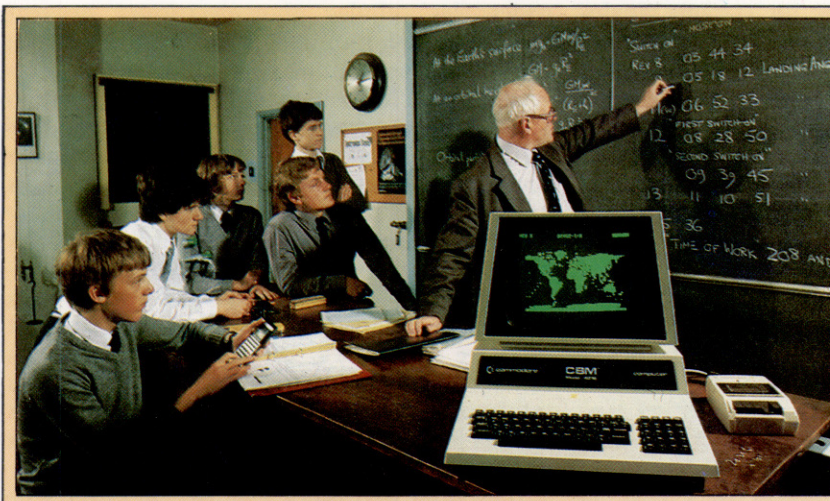


En raison de la rotation terrestre, les étoiles « glissent » hors du champ de vision en quelques minutes. Pendant l'utilisation d'un télescope qui n'explore qu'une étroite zone de ciel, il est essentiel qu'il puisse bouger continuellement pour compenser le mouvement terrestre. Des moteurs d'entraînement mécaniques sont utilisés depuis de nombreuses années, mais, depuis peu, l'amateur a accès à des systèmes commandés par ordinateur. Le télescope est monté sur un axe *équatorial* pointé vers le nord réel, et le moteur d'entraînement fait tourner l'axe selon la vitesse précise de rotation de la Terre, afin de maintenir l'objet dans le champ de vision. L'axe et le télescope sont munis d'encodeurs et de numériseurs qui envoient un signal numérique à l'ordinateur. De cette façon, un télescope peut suivre automatiquement pendant toute une nuit, sous la commande de l'ordinateur, un corps céleste sélectionné, pendant que son

Les ordinateurs peuvent également servir, en observation astronomique, à tenir compte des changements atmosphériques (par exemple, les variations de température et d'humidité qui réfractent et dévient la lumière dans son passage dans l'atmosphère).

L'astronomie n'a pas simplement utilisé les ordinateurs, elle a aussi contribué au développement de l'informatique. Le langage FORTH fut conçu par un astronome, Charles H. Moore, en 1971, à l'observatoire de Kitt Peak (Arizona), pour commander les radiotélescopes.

A l'intention des mordus de l'astronomie, des livres consacrés à la programmation domestique font leur apparition. Il existe même un magazine, nommé *Apex*, dans lequel des utilisateurs d'ordinateurs domestiques échangent et publient des programmes. Certains ont été écrits pour calculer le début des saisons, la conversion de dates historiques dans le calen-



Experts

Les élèves et le personnel de cette école anglaise ont acquis une excellente réputation en ce qui concerne leurs travaux en astronomie, spécialement dans le domaine de l'observation des satellites. A de nombreuses occasions, cette école fut la première station d'observation à détecter la présence d'un nouveau satellite en orbite. (Cl. John Drysdale/Colorific.)

image est lentement enregistrée sur une plaque. Certains astronomes amateurs américains ont même intégré les nouvelles technologies de reconnaissance et de synthèse de la parole dans leurs systèmes. Un ordinateur est programmé pour reconnaître certains mots de commande; ainsi, lorsqu'un observateur entre dans l'immeuble et prononce le mot « ouvrir », le dôme s'ouvre; un autre appel, « tourner le dôme », sollicite les moteurs et fait tourner le dôme sur ses roues. La synthèse de la parole est également très utile dans l'obscurité d'un observatoire pour émettre une information en provenance de l'ordinateur.

Les astronomes professionnels utilisent des télescopes qui enregistrent une partie du spectre électromagnétique située en dehors du domaine visible — comme les ondes radio ou les rayons X. En 1964, en Grande-Bretagne, le télescope Mark II de Jodrell Bank muni d'un disque elliptique de 38 m par 25 m, fut le premier télescope à utiliser un ordinateur numérique pour convertir les coordonnées célestes en des instructions destinées aux divers moteurs d'entraînement; ce processus de commande était effectué continuellement quatre fois par seconde.

drier julien, le lever et le coucher de la Lune, ou les tables quotidiennes destinées à déterminer de façon précise l'endroit du ciel où apparaîtra la comète de Halley en 1986.



Radiosources

Suite à la découverte d'objets émettant des ondes radio dans les lointaines galaxies, cet énorme radiotélescope fut construit à Jodrell Bank, au sud-ouest de Manchester. Les radiotélescopes, qui fonctionnent comme des énormes antennes, peuvent détecter des objets qui demeurent invisibles pour les plus puissants télescopes.

Étape par étape

Lorsqu'il est nécessaire de mesurer un mouvement linéaire ou angulaire au moyen d'un capteur optique, le code binaire n'est pas approprié. Le code Gray doit être utilisé.

Tableau de conversion

Ce tableau donne les équivalents des codes binaires et des codes Gray pour les nombres décimaux de 0 à 15.

Décimal	Binaire	Code Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Comment déterminer la position physique d'un objet mobile et la transmettre à l'ordinateur? Avec un robot, par exemple, l'ordinateur doit connaître les positions et les orientations de tous ses « membres »; dans une machine-outil commandée par ordinateur, la position de chaque intervention doit être calculée précisément. Comment une position peut-elle être convertie en une valeur binaire en vue d'un traitement informatique?

Il est possible d'utiliser un système analogique. Le dispositif mobile doit alors être connecté à une résistance variable et la tension résultante transmise à un convertisseur analogique-numérique (ou directement dans le port analogique si votre ordinateur en possède un). Cependant, cela ne constitue pas un système très précis; les pièces mécaniques sont sujettes à l'usure.

On peut aussi imprimer un code binaire sur le dispositif mobile et lire ce code directement dans l'ordinateur. Le code prend généralement la forme d'une configuration de blocs blancs et noirs imprimés. Ils sont lus au moyen d'une source lumineuse éclairant la configuration avec une ligne de cellules photoélectriques, chacune responsable d'un chiffre dans la configuration binaire. Lors d'un mouvement de l'outil, la configuration apparaissant sous les cellules

change et donne une sortie binaire qui définit la position du dispositif. En plus des arrangements linéaires, des configurations radiales sont utilisées pour coder les mouvements angulaires, comme celui d'une articulation d'un bras robot.

Des problèmes surviennent quand le dispositif se déplace d'un code binaire à un autre, et s'il s'immobilise entre deux codes. L'impression n'est précise que jusqu'à un certain seuil de tolérance. Lorsque le dispositif s'arrête entre deux codes, les cellules photoélectriques peuvent choisir de lire l'un des deux codes. Si l'outil s'arrête à un endroit où ses cellules lisent l'intersection entre la position 11 (1011) et la position 12 (1100), par exemple, le bit le plus significatif (le 1 à l'extrême gauche) est le seul qui puisse donner une valeur correcte, alors que les trois autres cellules proposent des valeurs contradictoires. Il existe certaines situations où tous les bits changent, comme la jonction entre les positions binaires 7 (0111) et 8 (1000). Ainsi, des imprécisions mineures d'impression pourraient produire des lectures incorrectes sur toutes les cellules. De quoi donner une valeur de position totalement fautive... L'ordinateur n'aurait aucune manière de déceler cette erreur, dont les conséquences pourraient être désastreuses.

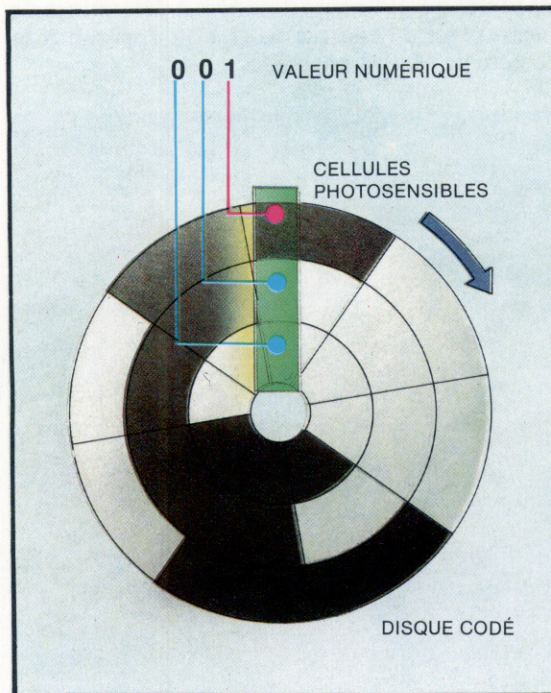
Il est donc nécessaire d'utiliser un système de comptage autre que binaire. Cela signifie que seul un bit peut être douteux sur toute jonction, et que l'erreur de sortie ne peut être que d'une position. Cet autre code se nomme le *code Gray* et sa caractéristique est la suivante : passer d'une valeur à une autre ne change qu'un seul bit, et ce doit être le bit le plus à droite qui donne toujours une configuration unique. Ainsi, si nous commençons par 0000, tout comme en binaire, le nombre 1 sera représenté par 0001. Cependant, pour représenter 2, nous devons changer le second bit pour obtenir 0011. Pour 3, il est maintenant possible de changer le premier bit, et d'obtenir 0010. Notez la différence avec la séquence binaire des mêmes nombres : 0000, 0001, 0010, 0011.

L'encadré illustre ce processus jusqu'au nombre décimal 15, en donnant les équivalents binaires à titre de référence. Comme exercice, vous pourriez essayer de trouver les codes Gray pour des valeurs plus grandes.

Les ordinateurs pourraient être conçus de façon à effectuer les calculs et à traiter les fonctions internes en code Gray; mais ce serait inefficace et inutile. On doit donc prévoir la conversion des codes Gray en codes binaires, par matériel ou par logiciel.

Angle de vision

La position angulaire d'un dispositif peut être lue par un ordinateur au moyen d'un disque sur lequel un code est imprimé. Un signal numérique parallèle est produit et modifié pendant le déplacement de l'objet. L'utilisation d'un code binaire pose un problème : si le disque s'arrête sur une jonction entre deux valeurs, un résultat non significatif peut être produit. Le code Gray résout ce problème. (Cl. Kevin Jones.)





Apple IIe

On se demande si l'Apple II n'est pas démodé. Mais cet ordinateur est toujours le plus polyvalent de tous les micro-ordinateurs, et il a des dispositifs complémentaires plus nombreux que tous les autres.

A plusieurs points de vue, l'Apple II fut un pré-curseur. Bien qu'il ne fût pas le premier micro-ordinateur, c'est à la suite du lancement de cette machine que des éléments comme la couleur, les graphiques haute résolution et le son intégré devinrent accessibles à l'utilisateur d'ordinateur domestique. Cependant, les caractéristiques vraiment significatives qui ont rendu (et continuent de le faire) l'Apple II si populaire ne sont pas évidentes à première vue.

Une des plus apparentes est la qualité de la documentation. Les manuels ont été préparés avec le souci de rendre chaque aspect de la machine aussi clair que possible. C'était en contradiction avec l'attitude des quelques sociétés qui proposaient déjà des matériels sur le marché et qui gardaient (et dans certains cas continuent à garder) un silence absolu concernant leurs produits. Cette accessibilité de l'information permit de rendre très connue une autre caractéristique de l'Apple II.

Il s'agit de la présence de connecteurs d'extension situés à l'arrière de la machine. C'est la souplesse d'organisation de ces connecteurs qui a rendu possible la conception de la vaste gamme de périphériques offerts pour cette machine. Cette diversité a ainsi permis d'utiliser l'Apple II de plusieurs façons.

Évidemment, de nombreux ordinateurs possèdent un bloc d'extension; mais généralement, il se limite à un ou deux connecteurs, et ceux-ci ne disposent que de très peu de mémoire. L'Apple II en a sept sur la version la plus récente (le IIe) et huit sur les modèles précédents (II et II Plus), chacun apparaissant à l'UC comme deux sections de mémoire (l'une assez petite, et l'autre une 2 K très pratique).

Par conséquent, une carte périphérique qui est branchée dans l'Apple peut contenir 2 048 octets de programmes de commande. L'utilisation d'une telle carte devient très simple, puisqu'il n'est pas nécessaire de l'associer à des programmes de commande spéciaux ou de réécrire le système d'exploitation.

Une vaste sélection de cartes est maintenant disponible pour l'Apple II, allant des interfaces d'entrée-sortie relativement simples aux cartes très évoluées, comme celles de crayon optique ou de mémoire RAM, qui peuvent étendre la mémoire jusqu'à 1 méga-octet ou plus. Cependant, puisque le 6502 ne peut adresser que 64 K, la mémoire est disposée par « banques » de 64 K, dont seulement une est sollicitée à la fois. Des ordinateurs munis de microprocesseurs 16 ou 32 bits peuvent même être branchés en paral-

lèle avec le processeur 6502 de la machine, pour créer un système aussi (ou plus) puissant qu'un mini-ordinateur.

Naturellement, une très vaste gamme de logiciels a été développée pour tirer profit de ce matériel. Aujourd'hui, la bibliothèque de programmes Apple est de loin la plus importante au monde — plus grande même que la liste de programmes qui utilisent le système d'exploitation standard CP/M. En fait, la bibliothèque CP/M peut être incluse dans la liste, car, si l'Apple II possède un processeur 6502, il peut exécuter les programmes CP/M à l'aide de l'un des dispositifs complémentaires les plus populaires, une carte Z80 (CP/M ne fonctionne que sur un microprocesseur Z80). Dès que cette carte est branchée dans l'un des connecteurs de l'Apple, la machine fonctionne comme un ordinateur CP/M parfaitement normal.

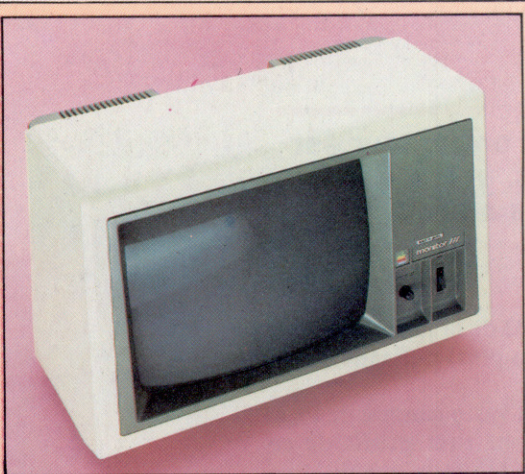
Bien qu'assez petit et d'apparence modeste, l'Apple II est une machine assez étonnante. C'est l'ordinateur domestique qui dispose du plus grand nombre de systèmes d'exploitation (11), de langages (au moins 27), et d'éditeurs de texte (12 ou plus).

L'évolution de l'Apple II est loin d'être terminée. Au cours de l'année 1984, un système d'exploitation entièrement nouveau sera lancé. Appelé ProDOS, il offre de nombreuses caractéristiques qui rendront l'Apple II supérieur sur le plan des performances à certains systèmes plus coûteux.

Le clavier de l'Apple

Le clavier a été conçu en visant les normes les plus élevées et en le destinant au traitement de texte. Les touches sont bien profilées et forment un arc de cercle qui facilite l'utilisation. Les deux touches portant le symbole de la pomme de chaque côté de la barre d'espacement sont des touches de commande utilisées dans des programmes d'application. A première vue, la touche RESET peut sembler être située dangereusement près de la touche DELETE. Cependant, pour remettre l'ordinateur à zéro, il est nécessaire d'appuyer simultanément sur les touches CTRL et RESET. (Cl. Chris Stevens.)





Moniteur

L'Apple IIe fonctionne avec tous moniteurs dédiés, mais celui proposé par Apple s'harmonise mieux avec la conception de l'Apple II. Il peut afficher jusqu'à 80 colonnes de texte. (Cl. Chris Stevens.)

UC 6502

Connecteur haut-parleur

Connecteur d'extension

Dans l'Apple IIe, le connecteur 3 est doublé par un connecteur plus important qui possède tous les signaux normaux du bus. Il intègre également la seconde mémoire de 64 K qui peut être ajoutée.

RAM

L'Apple IIe a une mémoire RAM de 64 K formée par 8 puces. La disposition actuelle de la RAM est plus complexe.

Mémoire vidéo

Les caractères inscrits sur l'écran sont créés par cette ROM qui permet l'utilisation de sept langages différents.

Mémoire clavier

La disposition du clavier est contrôlée par cette ROM qui s'adapte aux besoins des différents pays (clavier AZERTY en France et QWERTY en Allemagne par exemple).

Connecteur « souris »
L'Apple IIe peut recevoir ici un système auxiliaire d'entrée numérique des données.

Connecteur clavier

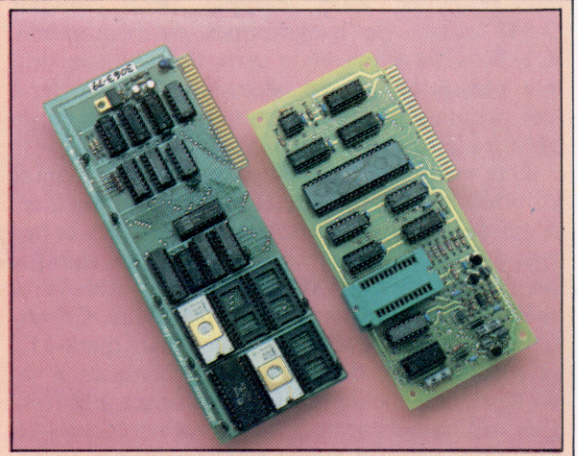


Lecteurs de disquettes

Les lecteurs de disquettes Apple ne sont pas « intelligents ». Une carte de contrôle doit donc être installée sur la carte principale de l'ordinateur. La capacité de 143 K est très basse, mais suffisante.

Réalisateur d'EPROM et carte ROM

La vaste gamme d'outils de développement offerts pour l'Apple facilite son utilisation comme système pour développer de nouveaux programmes. Le réalisateur d'EPROM peut être utilisé par quiconque pour produire des EPROM, qui peuvent être ensuite intégrés dans une carte ROM. Lorsque le programme est parfait, il peut être mis en ROM et vendu en série.





APPLE IIe

PRIX

DIMENSIONS

460 x 385 x 115 mm.

VITESSE DE L'HORLOGE

1 MHz.

MÉMOIRE

16 K de ROM.

64 K de RAM.

Extension jusqu'à 128 K ou plus par commutation de banques.

AFFICHAGE VIDÉO

24 lignes de 40 caractères monochromes seulement. Graphiques basse résolution de 48 x 40 en 16 couleurs. Graphique haute résolution 192 x 280 en 6 couleurs.

INTERFACES

Cassette, vidéo composite, 7 connecteurs d'extension, port de jeux.

LANGAGE INTÉGRÉ

BASIC.

AUTRES LANGAGES DISPONIBLES

La plupart des autres langages répandus, et certains langages assez rares.

ACCESSOIRES FOURNIS

Manuels d'installation et de BASIC, fil de téléviseur.

CLAVIER

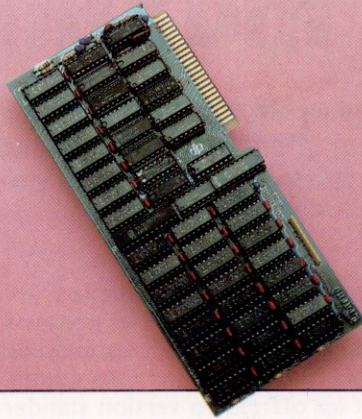
62 touches de haute qualité.

DOCUMENTATION

La documentation qui accompagne ce produit est excellente, bien que les manuels nécessaires pour comprendre entièrement la machine doivent être achetés séparément. Une vaste gamme de livres traitant tous les niveaux est disponible.

Carte RAM 256 K

Les fonctions de gestion de mémoire évoluées de l'Apple IIe permettent d'accéder par commutation à des banques de mémoire de 2 K, d'où un accès légèrement compliqué. Cependant, cela permet d'utiliser des blocs de RAM très grands. Cette carte donne une mémoire de 256 K, mais elle peut être portée à 1 million d'octets.



Unité de gestion mémoire

Une des deux ULA qui fait la différence entre l'Apple II et le IIe. Celle-ci contrôle les 80 colonnes de l'écran et la RAM.

Connecteur de puissance

Connecteur n° 1

L'imprimante se raccorde normalement ici.

Connecteur n° 7

Les effets spéciaux vidéo se font à partir de ce connecteur. C'est donc ici que l'on retrouve les crayons optiques par exemple.

Prise DIL pour jeux

Une des possibilités les plus attrayantes de l'Apple : une entrée analogique.

Entrée cassette

Sortie cassette

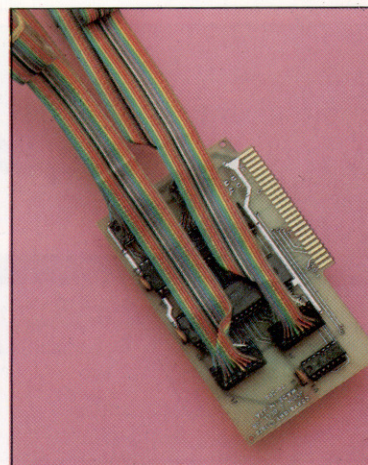
Sortie vidéo combinée

Connecteur auxiliaire vidéo

Unité d'entrée/sortie

Connecteur D pour jeux

La prise normale pour les jeux est trop fragile pour une utilisation quotidienne. Aussi l'Apple IIe dispose d'une petite prise D en parallèle.



Carte d'E/S universelle

Quelquefois, une carte peut avoir de si nombreuses fonctions qu'un contrôleur ROM limiterait l'utilisateur. Cette carte, qui a deux adaptateurs d'interface polyvalents, en est un exemple. Elle possède 40 lignes d'E/S à commande individuelle, deux registres à décalage qui servent à convertir les données parallèles en données série, et quatre rythmeurs 16 bits. (Cl. Chris Stevens.)



Chargement maximal

Les disques durs doivent fonctionner dans un environnement très pur. Le disque Winchester offre à l'utilisateur d'ordinateur domestique une capacité de stockage élevée et un temps d'accès très court.

Bien que la plupart des ordinateurs domestiques aient intégré de nombreuses caractéristiques des petits ordinateurs de gestion au cours des dernières années, un domaine de la technologie informatique est demeuré relativement peu évolué : le stockage de masse. Alors qu'un utilisateur d'ordinateur domestique est heureux de posséder un lecteur de disquettes capable de contenir 100 K de données, une machine de gestion nécessite un espace beaucoup plus grand. Une série de lecteurs de disquettes où serait éparpillée l'information ne permet pas de satisfaire ces besoins de grande capacité. Par conséquent, il fut nécessaire de développer des disques durs capables de stocker plus de données que des lecteurs de disquettes. Les premiers travaux sur les disques durs furent réalisés par IBM dans les années soixante. Les disques originaux contenaient 30 méga-octets sur chaque lecteur et ils fonctionnaient par paire. On les nomma donc 30/30, puis Winchester, à cause de la célèbre carabine 30/30!

Les disques Winchester emploient des plateaux rigides au lieu du plastique souple utilisé pour les disquettes. Cela permet de porter la densité de pistes de 96 TPI (*tracks per inch*) sur les disquettes à plusieurs centaines de TPI sur les disques Winchester. Grâce à une amélioration de la technique, on peut maintenant loger 10 ou même 20 méga-octets de données dans une boîte de même dimension qu'un lecteur de disquettes de 5 1/4 pouces.

Les utilisateurs d'ordinateurs domestiques commencent maintenant à profiter des avantages de cette technologie avec des produits comme les microlecteurs Sony 3 1/2 pouces et Hitachi 3 pouces. Il s'agit de lecteurs de disques semi-rigides, capables de stocker autant d'infor-

mations que toute disquette 5 1/4 pouces. Le BBC et l'Oric-1 sont parmi les premiers à avoir de tels dispositifs comme accessoires complémentaires, alors que des ordinateurs comme l'ACT Apricot les offrent en version standard.

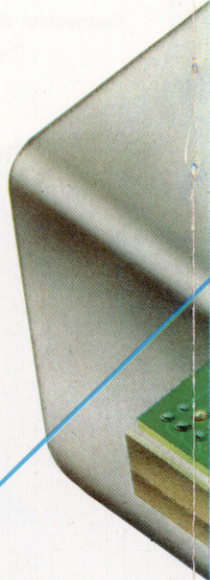
Cet accroissement de la densité de stockage a cependant créé d'autres problèmes. Par exemple, la précision demandée par le mécanisme a exigé la mise au point d'une méthode de déplacement de la tête entièrement nouvelle. La solution à ce problème fut trouvée dans l'industrie audio : les têtes Winchester sont souvent pilotées en position par une bobine électromagnétique semblable à celles qu'on trouve dans le haut-parleur. Un flux de courant dans la bobine crée un champ magnétique, et ce champ positionne de façon très précise une pièce métallique au centre de la bobine.

La tête « vole » au-dessus de la surface du disque sur un coussin d'air, et n'y touche jamais. L'usure des disques est fortement réduite mais ils doivent être scellés dans des boîtiers hermétiques pour éviter des problèmes causés par la poussière ou des corps étrangers. Généralement, le disque est fixé à l'intérieur du lecteur et ne peut être enlevé, bien que des disques Winchester munis de cartouches amovibles commencent maintenant à faire leur apparition. Ces cartouches sont scellées individuellement et ne sont ouvertes que lorsque la cartouche est insérée dans le lecteur pour permettre à la tête d'accéder à la surface du disque. Pour empêcher toute entrée de poussière, la pression d'air à l'intérieur de la cartouche est souvent plus grande qu'à l'extérieur.

Autre avantage des disques rigides : il est possible d'utiliser des plateaux multiples. Un disque Winchester de 10 méga-octets est simple-

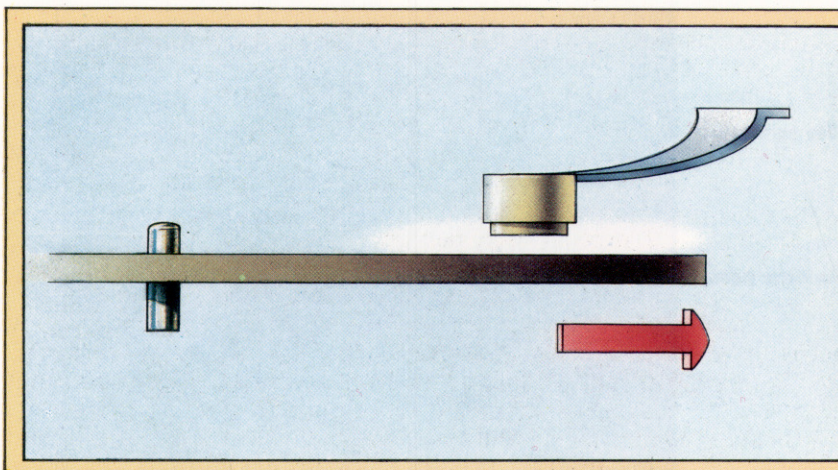
Boîtier

La plupart des disques Winchester sont contenus dans un boîtier d'alliage coulé sous pression. Cela permet d'aligner correctement les composants.



Plateaux

Les disques Winchester à grande capacité possèdent plusieurs plateaux ou disques sur le même axe de rotation. Ici, le disque en a cinq, mais la plupart en possèdent deux ou trois. Les têtes de lecture/écriture sont connectées ensemble; ainsi un seul bloc peut être lu à la fois.



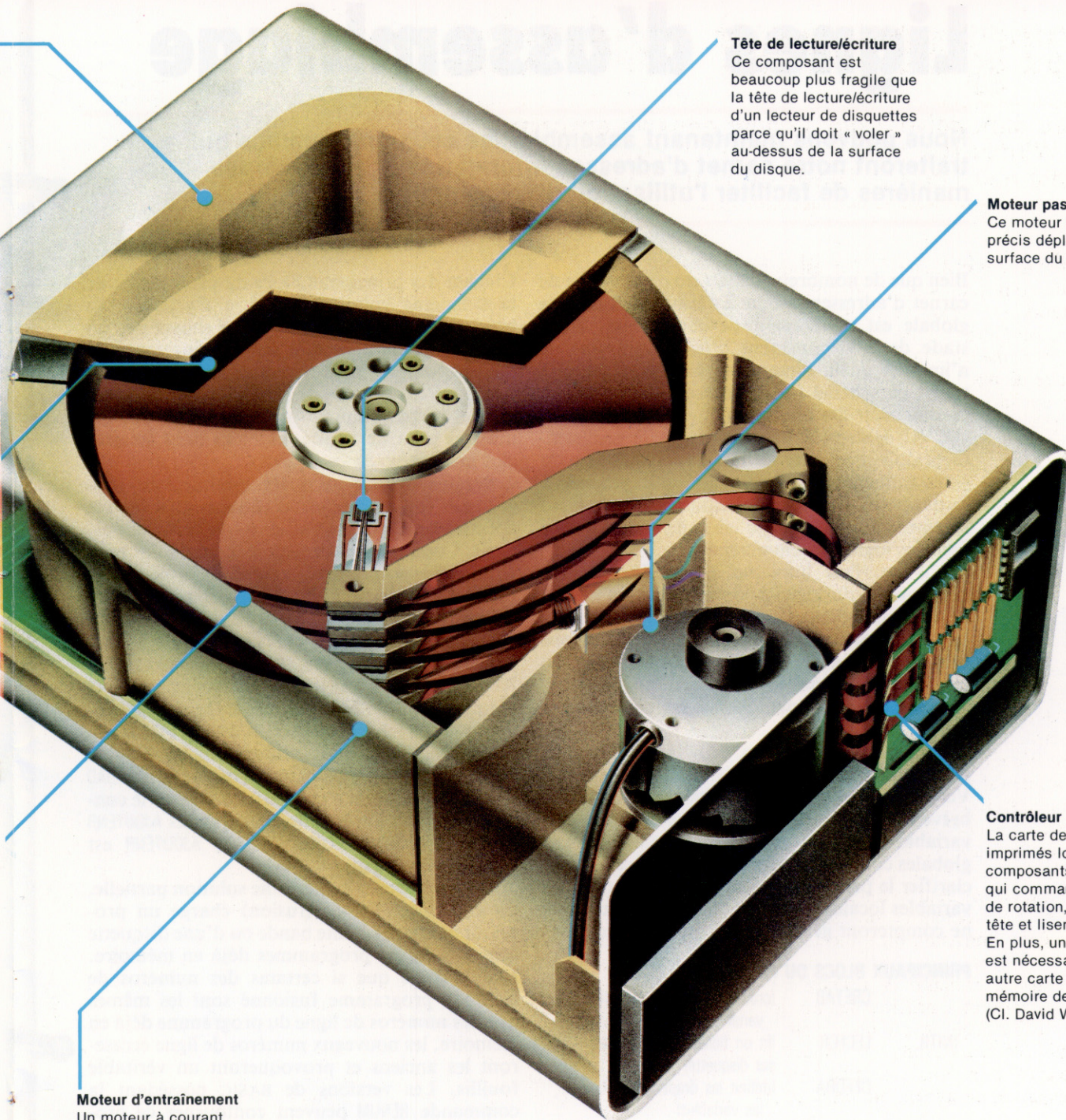
Têtes flottantes

Contrairement à un lecteur de disquettes où la tête glisse sur la surface d'enregistrement de la disquette, sur un lecteur Winchester, la tête « vole » très près de la surface du disque. Le disque tourne si rapidement qu'il crée un coussin d'air qui supporte la tête. Si la tête frotait le disque, elle enlèverait probablement la surface d'enregistrement magnétique.

(Cl. Kevin Jones.)

Scellage hermétique

Les pièces mécaniques du lecteur sont complètement scellées à l'abri de l'atmosphère, afin d'éviter que des particules de poussière ou de fumée n'encrassent la tête.

**Tête de lecture/écriture**

Ce composant est beaucoup plus fragile que la tête de lecture/écriture d'un lecteur de disquettes parce qu'il doit « voler » au-dessus de la surface du disque.

Moteur pas à pas

Ce moteur électrique très précis déplace la tête sur la surface du disque.

Contrôleur

La carte de circuits imprimés loge les composants électroniques qui commandent la vitesse de rotation, positionnent la tête et lisent les données. En plus, un DOS sophistiqué est nécessaire, soit sur une autre carte ou dans la mémoire de l'ordinateur. (Cl. David Weeks.)

Moteur d'entraînement

Un moteur à courant continu et un petit générateur sont montés sur le même axe qui fait tourner le disque. La sortie du générateur correspond à la vitesse du moteur et est retransmise vers un circuit spécial de commande. C'est de cette façon que la vitesse est déterminée aussi précisément.

ment construit à partir de deux plateaux de 5 méga-octets placés dans le même boîtier. La gestion de tout cet espace de stockage est simplement assurée en le divisant en de nombreuses sections. Pour maintenir la compatibilité avec les logiciels existants, qui ne prévoient que d'utiliser des données sur des disquettes, ces sections ont généralement une capacité qui est proche de celle des disquettes. Un disque Winchester apparaît pour l'ordinateur comme plusieurs unités de disquettes.

Quand un disque Winchester est formaté (c'est-à-dire lorsque les pistes et les secteurs sont marqués), le DOS doit être capable de reconnaître les « mauvais secteurs » (surfaces d'enregistrement défectueuses). Sur une disquette, un

mauvais secteur signifie une disquette inutilisable; sur un disque Winchester, le formatage ne fait que noter que le secteur est défectueux et l'invalidé pour toute utilisation ultérieure. Après tout, quelques centaines d'octets sont une quantité négligeable sur une capacité supérieure à 5 millions d'octets.

Évoluant au même rythme que le reste de la technologie informatique, les disques Winchester deviennent plus petits; mais il existe déjà un microdisque Winchester de 5 millions d'octets. Avec des développements aussi rapides dans les unités de stockage sur disque, il est possible que des constructeurs puissent bientôt proposer des lecteurs de disques durs pour ordinateurs domestiques à un prix abordable.

Lignes d'assemblage

Nous pouvons maintenant assembler les sous-programmes qui traiteront notre carnet d'adresses informatisé et examiner les manières de faciliter l'utilisation du programme.

Bien que de nombreux détails du programme de carnet d'adresses restent à définir, la structure globale est maintenant devenue claire. A ce stade de développement d'un programme de n'importe quelle dimension, il est préférable de tracer un schéma fonctionnel et de réfléchir au déroulement des opérations.

C'est également à ce stade que le programmeur doit considérer l'« interface humaine » du programme ainsi que l'image qu'il propose à l'utilisateur. Ces notions importantes reçoivent rarement l'attention qu'elles méritent, même dans les logiciels commerciaux.

L'« interface humaine » signifie, en fait, l'ergonomie du logiciel, ou sa facilité d'utilisation. L'image proposée à l'utilisateur est la façon dont il perçoit le programme utilisé. Nous devons examiner ces notions par rapport au programme que nous avons développé jusqu'ici et déterminer comment les mettre en œuvre au mieux.

Le tableau illustre les principaux blocs du programme qui ont été examinés jusqu'ici. Voici donc les noms de ces blocs, suivis d'une brève description de leur fonction. Les noms de variables de chaînes et des variables numériques globales compteront plusieurs caractères afin de clarifier le programme, tandis que les noms des variables locales (dans des boucles par exemple) ne compteront généralement qu'un caractère.

PRINCIPAUX BLOCS DU PROGRAMME

	CRETAB	(crée des tableaux et initialise des variables)
INITIL	LECFRC	(lit un fichier sur bande ou sur disquette)
	DEFDRA	(définit les drapeaux et modifie les variables)
ACCUEILLE		(affiche un message d'accueil)
CHOIX	CHMENU	(affiche le menu de choix)
	CHOI	(affecte l'option à CHOI)
	TROUVENR	(trouve et imprime un enregistrement)
	TROUVNOM	(trouve des noms à partir d'une entrée partielle)
	TROUVIL	(trouve les enregistrements d'une ville spécifique)
	TROUVINT	(localise des noms à partir d'initiales)
EXECUT	LSTENR	(liste les enregistrements)
	AJOUTENR	(ajoute un enregistrement)
	MODENR	(modifie un enregistrement)
	EFFENR	(efface un enregistrement)
	QUITTE	(sauvegarde le fichier et quitte le programme)

Chacun des principaux blocs du programme de la deuxième colonne doit être fractionné en sous-unités. Celles-ci doivent être encore perfectionnées jusqu'à ce que nous ayons assez de détails pour écrire le véritable programme en BASIC. Les processus impliqués dans cette forme de programmation progressive ont été illustrés pour de nombreux blocs dans les articles précédents de ce cours.

Supposons que tous les modules du programme (ou la plupart d'entre eux) aient été définis, codés en BASIC et testés individuellement. Comment peuvent-ils être assemblés pour former un programme complet? La meilleure façon d'aborder ce problème est de sauvegarder chaque module sur bande ou sur disquette, en leur donnant les noms choisis lors du développement du programme. Par conséquent, AJOUTENR peut être écrit et testé, puis sauvegardé sous le nom AJOUTENR. Normalement, lorsque le programme est chargé à partir d'une bande ou d'une disquette, nous utilisons la commande LOAD, suivi du nom de fichier, comme ceci : LOAD AJOUTENR. Mais cela a pour effet d'effacer le contenu de la mémoire; si nous chargeons AJOUTENR puis chargeons ensuite MODENR, AJOUTENR est effacé entièrement.

Heureusement, il existe une solution partielle. La commande MERGE (fusion) charge un programme à partir d'une bande ou d'une disquette sans effacer les programmes déjà en mémoire. Mais notons que si certains des numéros de ligne du programme fusionné sont les mêmes que des numéros de ligne du programme déjà en mémoire, les nouveaux numéros de ligne écraseront les anciens et provoqueront un véritable fouillis. Les versions de BASIC possédant la commande RENUM peuvent contourner ce problème en renumérotant les lignes dans un module de programme avant la sauvegarde.

Malheureusement, de nombreuses versions ne possèdent pas la commande RENUM. Il est donc nécessaire de planifier soigneusement les numéros de ligne dès le début. Lorsqu'un schéma complet des modules principaux du programme a été tracé (comme nous l'avons fait partiellement dans le tableau), il est possible de définir les numéros de ligne de départ de chaque bloc. Certaines parties du programme susceptibles de subir d'importantes modifications, comme le programme principal ou les parties de gestion de fichiers du programme, doivent avoir des incréments de 50, voire 100, pour laisser assez d'espace aux modifications. Les modules du programme moins exposés aux modifications,

comme la routine ACCUEIL, peuvent avoir des incréments de ligne de 10. L'insertion de nombreuses lignes REM dans le programme ne fait pas que faciliter la lecture du programme; elle permet également l'addition ultérieure d'instructions ou d'appels de sous-programmes. Si en plus votre version ne possède pas la commande MERGE, vous devrez taper les divers modules et les sauvegarder tous ensemble.

Le premier bloc du programme principal est INITIL, qui est censé initialiser des variables, déclarer des tableaux, lire des fichiers, affecter des données aux tableaux, définir des drapeaux, etc.

Le sous-programme INITIL peut être fractionné en *CRETAB* (créer des tableaux), *LECFER* (lire des fichiers et affecter des données aux tableaux appropriés) et *DEFDRA* (définir des drapeaux).

Lorsque cela a été fait, le programme passe à *ACCUEIL*, un sous-programme qui affiche un message à l'écran. La dernière partie de ce sous-programme attend que l'utilisateur presse la barre d'espace pour continuer l'exécution.

Le programme passe alors à *CHOIX*, qui comprend deux parties : la première présente un menu des options offertes par le programme de carnet d'adresses, la seconde accepte l'entrée du choix donné au clavier et affecte la valeur (numérique) à une variable nommée CHOI.

La valeur de cette variable est utilisée par le bloc suivant du programme, EXECUT, afin de sélectionner un des neuf autres blocs du programme. Tous, sauf QUITTE, doivent pouvoir revenir à *CHOIX* après avoir été exécutés, car l'utilisateur doit pouvoir sélectionner une autre option. Ce ne sera pas nécessaire si l'option 9 (QUITTE) a été sélectionnée, parce qu'elle est censée terminer le fonctionnement du programme.

Le principal problème de ce programme se situe au niveau de la séquence d'exécution. INITIL insiste pour que nous lisions un fichier dans un dispositif de stockage de masse, que le fichier existe ou non. Si le programme est exécuté pour la première fois, en l'absence d'enregistrement, toute tentative pour ouvrir ou lire un fichier non existant donnera un message d'erreur et le programme ne fonctionnera pas.

Il est donc nécessaire d'appeler la routine *LECFER* uniquement à partir de l'un des modules EXECUT, et seulement une fois lors de chaque exécution du programme. Cela implique la présence obligatoire d'un drapeau ENTDRA, mis à 0 au départ, puis mis à 1 dès que le fichier a été lu. AJOUTENR recherche alors dans les tableaux pour localiser le premier élément vide et écrit l'information à cet endroit. Cet enregistrement ne sera certainement pas dans la bonne séquence de tri. Il doit donc y avoir un drapeau RMOD qui est mis à 1 lors d'une exécution. Le drapeau RMOD doit aussi être mis à 1 si MODENR ou EFFENR sont exécutés. Vous pouvez essayer d'écrire le code approprié ou, si vous désirez simplement exécuter le programme, changez la ligne 1310 en RETURN.

Des opérations telles qu'ajouter, effacer ou modifier un enregistrement impliquent que la séquence d'enregistrement risque de ne plus être

en ordre. Tout module (TROUVENR par exemple) doit d'abord vérifier RMOD pour voir si des changements ont été apportés. Si c'est le cas, le programme doit faire un tri avant d'effectuer la recherche, ou effectuer une recherche inefficace dans une pile. QUITTE vérifie automatiquement RMOD et appelle la routine de tri si RMOD est mis à 1 avant de sauvegarder les données dans le fichier sur bande ou sur disquette.

Les aspects de l'« interface humaine » que nous avons mentionnés précédemment peuvent comprendre les catégories suivantes :

- l'interface utilisateur;
- l'image proposée à l'utilisateur;
- la reprise d'erreur;
- la sécurité;
- la possibilité d'adaptation.

Nous avons opté pour l'utilisation de menus (plutôt que de commandes). De nombreuses personnes préfèrent des commandes, mais le point important est que, quelle que soit la communication utilisée, le mode d'interaction choisi doit être constant. Des commandes similaires ne doivent pas exécuter des opérations différentes dans diverses parties du programme. Si de telles commandes existent, l'utilisateur doit lire soigneusement chaque menu avant de faire un choix et ne peut développer de réflexes d'utilisation pour accélérer son travail.

Tel quel, notre programme est médiocre : le message d'accueil disparaît en appuyant sur la barre d'espace; le menu des options disparaît automatiquement en tapant l'une des touches numériques 1 à 9; et l'entrée de données dans AJOUTENR se termine (pour chaque champ) en appuyant sur la touche RETURN. Ce type d'inconstance peut être acceptable dans un programme d'amateur, mais est inadmissible dans un logiciel commercial.

L'image proposée à l'utilisateur correspond à la manière dont celui-ci perçoit le fonctionnement du programme. La plupart des opérations effectuées à l'intérieur de l'ordinateur se font entièrement à l'insu de l'opérateur. La seule façon de permettre à l'opérateur d'être conscient des opérations du programme est de lui proposer divers affichages associés à ces opérations. L'image que nous désirons proposer à l'utilisateur est un réel carnet d'adresses. De façon similaire, l'image qu'un programme de traitement de texte devrait proposer à l'utilisateur serait une feuille de « papier » à l'écran. En principe, à l'écran, les caractères gras devraient apparaître gras, les caractères soulignés devraient l'être, et le texte justifié (tapé avec une marge droite alignée) devrait l'être également.

L'image proposée à l'utilisateur est rarement parfaite — aucun carnet d'adresses réel ne vous demande PRESSER 1 pour trouver un nom. Néanmoins, un bon programme implique la présence d'affichages bien conçus et une disposition constante des opérations. Les messages doivent toujours apparaître à la même position sur l'écran (certains programmes de traitement de texte affichent par exemple des messages en

haut de l'écran et d'autres messages en bas, apparemment au hasard). Un programme bien conçu informera également l'utilisateur de sa position dans le programme à tout moment. Si vous êtes dans le mode AJOUTENR, il devrait y avoir un message pour vous indiquer que vous vous trouvez à ce niveau du programme. Si vous venez d'entrer un champ (pour ajouter un enregistrement), le programme devrait afficher un message comme celui-ci : APPUYEZ SUR RETURN SI L'ENTRÉE EST CORRECTE, SINON APPUYEZ SUR ESCAPE (ce qui nous conduit à l'important sujet de la reprise d'erreur et de la signalisation, sur lequel nous reviendrons).

Tous les formats devraient apparaître à l'écran, afin que, par exemple, l'enregistrement affiché sur l'écran soit du même format que celui qui est imprimé par l'imprimante. De nombreux logiciels commerciaux comportent des menus d'aide qui vous indiquent que faire si vous êtes dans le doute.

L'image proposée à l'utilisateur par un programme est meilleure concrète — une feuille de papier ou une fiche — qu'abstraite avec des sous-fichiers, des tampons, etc. De nombreux programmes de base de données ont des lacunes à cet égard ; l'utilisateur doit constamment garder à l'esprit le fait que certains éléments d'information sont dans des sous-fichiers ou dans des champs cachés, temporaires. Ces caractéristiques ont tendance à rendre assez désagréable l'utilisation d'un programme.

La reprise d'erreur est un autre sujet très important. Que se passe-t-il, par exemple, si en entrant un nom vous vous rendez compte que vous avez fait une erreur ? Devez-vous continuer et appeler ensuite MODENR pour corriger cette erreur, ou le programme vous donne-t-il l'occasion de « sortir » avant de continuer ? La plupart des versions de BASIC signalent les erreurs de saisie d'un programme, soit lors de l'entrée de la ligne erronée, soit lors de l'exécution du programme. Le BASIC offre de nombreux messages qui demandent à l'utilisateur d'entrer de nouveau une entrée si celle qui est fournie par l'utilisateur est incorrecte (par exemple, le message REDO qui est produit lorsque l'entrée fournie en réponse à une instruction INPUT est inadéquate).

La gestion des erreurs a deux facettes : leur signalisation et leur reprise. Par exemple, un programme de traitement de texte bien connu a une excellente signalisation d'erreur, mais de très mauvaises possibilités de reprise ; si vous créez un long document et si vous essayez de le sauvegarder, le programme vous donne le message suivant : DISQUETTE PLEINE. Il ne permet donc pas de réparer l'erreur, et une nouvelle disquette ne peut être formatée sans détruire d'abord le texte que vous avez mis des heures à taper !

Toute opération effectuée par l'utilisateur qui pourrait entraîner la perte de données devrait faire l'objet d'une confirmation avant d'être exécutée. Des messages comme CECI DÉTRUIRA LES ENREGISTREMENTS. CONFIRMATION ? (O/N) devraient toujours être prévus. Dans un programme de

traitement de texte, le message suivant pourrait être produit : LA COMMANDE « SAUVEGARDE » NE GARDERA PAS UNE COPIE DE L'ANCIEN DOCUMENT. CONFIRMATION ? (O/N).

Le traitement des erreurs (signalisation et reprise) devrait être envisagé, dès la conception du programme, à tout endroit où il y a possibilité d'entrée de mauvaises données, mauvais choix du menu, mauvaises commandes et lorsque les données doivent être modifiées et sauvegardées (surtout si l'opération implique la destruction des anciennes données).

Vous devez accorder de l'importance à la sécurité — que se passe-t-il au niveau du programme et des données en cas d'erreur fatale (comme une panne de courant) ? Le concepteur du programme doit examiner combien de données il est possible de perdre et concevoir des méthodes pour en récupérer le plus possible. Un autre programme de traitement de texte sophistiqué comporte un programme nommé RECOVER qui permet, lors d'une erreur catastrophique (comme une panne de courant, ou la mise hors tension de l'ordinateur avant la sauvegarde du document) de ne presque rien perdre. De telles techniques de programmation évoluées sont malheureusement hors des objectifs de ce cours. Rappelons cependant qu'il est important, d'une part, d'anticiper toutes les erreurs fatales que vous êtes en mesure de traiter et, d'autre part, d'écrire des routines pour les gérer.

La possibilité d'adaptation, la facilité de personnaliser un programme sont aussi importantes. Nous avons déjà abordé ce sujet à plusieurs reprises. Laissez d'abord beaucoup d'espace entre les numéros de ligne (en BASIC) et insérez beaucoup d'instructions REM vides qui pourront plus tard être remplacées par des instructions ou des GOSUB. Lors de la création des tableaux, prévoyez au moins un tableau redondant en vue d'une extension ultérieure. Un bon programme peut toujours être amélioré. Pour ce faire, il est nécessaire d'écrire plus d'instructions.

Variantes de basic



Voir le programme donné à la page 318 pour la version du Spectrum des lignes 1300 à 1370. La boucle du programme principal entre les lignes 3750 et 3780 fonctionnera, mais pourra causer des problèmes car les touches du Spectrum sont répétées si elles sont maintenues enfoncées pendant plus d'une fraction de seconde. La notice recommande que ce type de boucle INKEY\$ renferme une ligne supplémentaire pour l'éviter :

```
3755 IF INKEY$ < > THEN GOTO 3755
```

Le Spectrum gère la fonction VAL(A\$), mais le programme sera interrompu si le premier caractère de A\$ est non numérique ; dans ce programme le problème peut être évité avec :

```
3780 LET CHOI = CODE A$:48
```

mais ce n'est pas une solution complète — elle ne fonctionne que lorsque A\$ est un simple caractère (comme ce doit être dans le programme). Le Spectrum ne peut



MERGE

utiliser ON...GOSUB, mais vous permet d'écrire GOSUB (expression numérique) ainsi que simplement GOSUB (numéro de ligne); la ligne 4010 peut donc être remplacée par :
4010 GOSUB (290 + CHOI*20)

Le Spectrum n'offre pas la commande RENUM.

Cette commande n'est pas offerte sur l'Oric-1, le Commodore 64, le Vic-20, le Lynx, le Dragon 32 et le BBC. Souvent cependant, il existe une façon de simuler la commande MERGE que des groupes d'utilisateurs ou de publications spécialisées révèlent. Le Lynx offre la commande APPEND qui vous permet de charger un programme à la fin du programme en mémoire; cela peut remplacer MERGE à la condition que le premier numéro de ligne du programme sur cassette soit plus grand que le dernier numéro de ligne du programme en mémoire.

RENUMBER

INKEYS

OPEN CLOSE

Cette commande n'est pas offerte par le Spectrum, les Commodore et l'Oric-1 : il est possible de renuméroter ligne par ligne, sur ces machines, en utilisant les fonctions d'édition.

Voir « Variantes de basic » précédents.

PRINT CHR\$(12)

Remplacez par CLS, sauf sur le Commodore 64 et sur le Vic-20; sur ces machines tapez PRINT (SHIFT) + la touche CLR, ce qui devrait faire apparaître un cœur inversé entre guillemets.

ON...GOSUB

A la ligne 4010, ON...GOSUB renvoie à des lignes inexistantes — 310, 330, 350, etc. — ce qui cause l'interruption du programme. Ces lignes seront écrites plus tard pour contenir les sous-programmes d'exécution du menu, en attendant, entrez ces lignes factices dans votre programme :

```
310 RETURN
330 RETURN
et ainsi de suite.
```

STR\$

Le Lynx n'offre pas cette commande, remplacez donc la ligne 9080 par :

```
9075 N = TAILLE
9077 GOSUB 9500
9080 NDXCHP$(TAILLE) = N$
et insérez ce sous-programme :
9500 REM POUR CONVERTIR N EN N$ OU N EST UN ENTIER
9510 LET N$ = « »
9520 IF N < 0 THEN LET N$ = « »
9530 N = ABS(N)
9540 X = 10
9550 I = 1
9560 FOR K = 1 TO 8
9570 I = I*X
9580 R = K
9590 IF I > N THEN K = 8
9600 NEXT K
9610 FOR K = 1 TO R
9620 I = I/X
9630 Z = INT(I/I)
9640 LET N = N - Z*I
9650 N$ = N$ + CHR$(48 + INT(Z))
9660 NEXT K
9670 RETURN
```

```
10 REM « PROGRAMME PRINCIPAL »
20 REM *INITIL*
30 GOSUB 1000
40 REM *ACCUEIL*
50 GOSUB 3000
60 REM *CHOIX*
70 GOSUB 3500
80 REM *EXECUT*
90 GOSUB 4000
100 END
1000 REM SOUS-PROGRAMME *INITIL*
1010 GOSUB 1100 : REM SOUS-PROGRAMME *CRETAB* (CRÉER DES TABLEUX)
1020 GOSUB 1300 : REM SOUS-PROGRAMME *LECFER* (LECTURE DE FICHIERS)
1030 GOSUB 1320 : REM SOUS-PROGRAMME *DEFDRA* (DÉFINITION DE DRAPEUX)
1040 REM
1050 REM
1060 REM
1070 REM
1080 REM
1090 RETURN
1100 REM SOUS-PROGRAMME *CRETAB* (CRÉER DES TABLEUX)
1110 DIM NOMCHP$(50)
1120 DIM MODCHP$(50)
1130 DIM RUECHP$(50)
1140 DIM CPOCHP$(50)
1150 DIM VILCHP$(50)
1160 DIM TELCHP$(50)
1170 DIM NDXCHP$(50)
1180 REM
1190 REM
1200 REM
1210 LET TAILLE = 0
1220 LET RMOD = 0
1230 LET SVED = 0
1240 LET ACT = 0
1250 REM
1260 REM
1270 REM
1280 REM
1290 RETURN
1300 REM SOUS-PROGRAMME *LECFER*
1310 ON ERROR GOTO 1370
1320 OPEN « I », #1, « ADBK.DAT »
1330 FOR L = 1 TO 50
1340 INPUT #1 NOMCHP$(L), RUECHP$(L), CPOCHP$(L), VILCHP$(L), TELCHP$(L)
1350 NEXT L
1360 CLOSE #1
1370 RETURN
1380 REM SOUS-PROGRAMME FACTICE *DEFDRA*
1390 RETURN
3000 REM SOUS-PROGRAMME *ACCUEIL*
3010 PRINT
3020 PRINT
3030 PRINT
3040 PRINT
3050 PRINT TAB(12);« *VOICI LE PROGRAMME* »
3060 PRINT TAB(4);« *DE CARNET D'ADRESSES INFORMATISÉ* »
3070 PRINT TAB(10);« *D'ABC INFORMATIQUE* »
3080 PRINT
3090 PRINT TAB(5);« (PRESSEZ LA BARRE D'ESPACEMENT POUR CONTINUER) »
3100 FOR L = 1 TO 1
3110 IF INKEY$ < > « » THEN L = 0
3120 NEXT L
3130 PRINT CHR$(12)
3140 RETURN
3500 REM SOUS-PROGRAMME *CHOIX*
3510 REM
3520 REM *CHMENU*
3530 PRINT CHR$(12)
3540 PRINT « PRENDRE UN DES SUIVANTS »
3550 PRINT
3560 PRINT
3570 PRINT
3580 PRINT « 1. TROUVER UN ENREGISTREMENT (A PARTIR D'UN NOM) »
3590 PRINT « 2. TROUVER DES NOMS (A PARTIR D'UN NOM INCOMPLET) »
3600 PRINT « 3. TROUVER DES ENREGISTREMENTS (A PARTIR D'UNE VILLE) »
3610 PRINT « 4. TROUVER DES ENREGISTREMENTS (A PARTIR D'INITIALES) »
3620 PRINT « 5. LISTER TOUS LES ENREGISTREMENTS »
3630 PRINT « 6. AJOUTER UN NOUVEL ENREGISTREMENT »
3640 PRINT « 7. CHANGER UN ENREGISTREMENT »
3650 PRINT « 8. EFFACER UN ENREGISTREMENT »
3660 PRINT « 9. QUITTER ET SAUVEGARDER »
3670 PRINT
3680 PRINT
3690 REM *ENTCHOI*
3700 REM
3710 LET L = 0
3720 LET I = 0
3730 FOR L = 1 TO 1
3740 PRINT « ENTREZ UN CHOIX (1-9) »
3750 FOR I = 1 TO 1
3760 LET A$ = INKEY$
3770 IF A$ = « » THEN I = 0
3780 NEXT I
3790 LET CHOI = VAL(A$)
3800 IF CHOI < 1 THEN L = 0
3810 IF CHOI > 9 THEN L = 0
3820 NEXT L
3830 RETURN
4000 REM SOUS-PROGRAMME *EXECUT* — VOIR VARIANTES DE BASIC
4010 ON CHOI GOSUB 310, 330, 350, 370, 390, 410, 430, 450, 470
4020 RETURN
9000 REM SOUS-PROGRAMME *AJOUTENR*
9010 PRINT CHR$(12)
9020 INPUT « ENTREZ LE NOM »; NOMCHP$(TAILLE)
9030 INPUT « ENTREZ LA RUE »; RUECHP$(TAILLE)
9040 INPUT « ENTREZ LE CODE POSTAL »; CPOCHP$(TAILLE)
9050 INPUT « ENTREZ LA VILLE »; VILCHP$(TAILLE)
9060 INPUT « ENTREZ LE NUMÉRO DE TEL. »; TELCHP$(TAILLE)
9070 LET RMOD = 1
9080 LET NDXCHP$(TAILLE) = STR$(TAILLE)
9090 GOSUB *MODNOM*
9100 RETURN
```





Qualités sonores

Production de sons sur le BBC modèle B.

Un potentiel sonore exceptionnel et des commandes BASIC complètes placent le BBC parmi les meilleurs ordinateurs domestiques offerts à l'utilisateur spécialement intéressé par le son. Trois oscillateurs indépendants, huit types de bruits et quatre générateurs d'enveloppes sont offerts en version standard. Cela signifie qu'il est possible de construire des séquences musicales impliquant trois voix et d'utiliser des commandes spéciales pour garantir que les notes sélectionnées pour former un accord soient jouées exactement au même moment.

Création sonore

Dans sa forme la plus simple, un son peut être créé à l'aide de la commande SOUND :

SOUND C, V, N, D

Dans ce cas :

- C = numéro du canal ou de l'oscillateur (0-3)
- V = volume
- N = note
- D = durée de la note

Un des trois oscillateurs (1, 2 et 3) peut être sélectionné pour produire une note, et 0 sélectionne le bruit. Le volume a une valeur com-

prise entre 0 (nul) et - 15 (fort). Les notes sont définies en quarts de tons entre 0 (*la* # à 116,5 Hz) et 255 (*ré* à 4 698,64 Hz), soit une étendue de cinq octaves et demie. Le *do* de référence a une valeur de 28. Si le canal de bruit a été spécifié, huit types de bruit sont disponibles de la façon suivante :

Numéro	Type de bruit
0	Trémolo aigu
1	Trémolo moyen
2	Trémolo grave
3	Trémolo. Le registre varie selon le registre du canal 1
4	Aigu
5	Moyen
6	Grave
7	Registre varie selon le registre du canal 1

Finalement, la durée d'une note est commandée par pas de 1/20 de seconde entre 1 et 255, soit une durée maximale de 12,75 secondes. Voici une commande qui produira le *la* au-dessus du *do* de référence sur le canal 1 à un volume moyen de - 7 pendant une demi-seconde :

SOUND 1, - 7, 46, 10

Si l'ordinateur reçoit une seconde commande SOUND avant la fin du premier son, la note est placée dans la file d'attente du canal.

Jeu de lumière

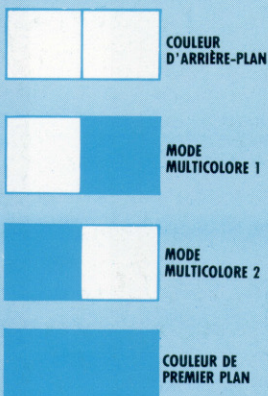
Capacités graphiques du Commodore 64.

Le Commodore 64 offre beaucoup de possibilités au programmeur graphique. Le principal inconvénient de cette machine, comme avec le Vic-20, est le jeu limité de commandes standard de BASIC. Lire et écrire (PEEK et POKE) dans la mémoire de l'ordinateur permettent au programmeur d'exploiter toutes les caractéristiques de la machine, mais certaines des procédures nécessaires peuvent être assez déroutantes. Avec le Vic-20, il existe plusieurs façons de contourner ce problème : des logiciels, produits par des fabricants indépendants, rendent beaucoup plus simple la création de plans-objets ou de caractères définis par l'utilisateur. En plus de ceux-ci, Commodore produit sa propre cartouche, nommée « Simon's BASIC ».

Le Commodore 64 possède les jeux de caractères minuscules et majuscules standard dans les modes d'affichage normal et inverse. Il offre aussi des caractères graphiques spéciaux, d'abord développés sur le PET puis sur le Vic-20. L'affichage (sur 40 colonnes et 25 lignes) des caractères s'obtient en les imprimant (PRINT) à l'écran ou en écrivant (POKE) les codes associés dans les mémoires écran et couleur décrites en détail dans le manuel de l'utilisateur. Les caractères graphiques spéciaux de Commodore représentent un outil très pratique et très souple pour composer des affichages intéressants en basse résolution.

Si le caractère désiré ne peut être trouvé, il est possible d'en définir de nouveaux pour composer l'affichage. Cela n'est malheureusement pas facile à faire. Pour définir un caractère, le programmeur doit d'abord copier le jeu de caractères normaux qui seront nécessaires dans la RAM à partir de la ROM, avant d'écrire les nombres nécessaires dans la série d'emplacements où doit se trouver le nouveau caractère.

Chaque cellule de caractères consiste en 8 lignes de 8 points qui sont représentées sous forme binaire. Normalement, 1 est interprété comme un point du premier plan ou de la couleur du caractère, et 0 est interprété comme un





Leonardo Torres

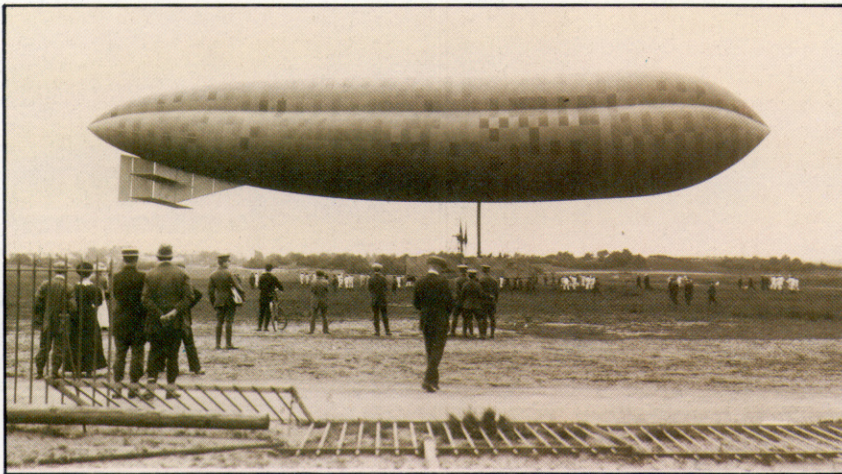
Il s'intéressa aux dirigeables et aux téléphériques, mais il apporta aussi une importante contribution au développement de l'informatique.

Leonardo Torres Quevedo, le premier scientifique qui a appliqué l'arithmétique à virgule flottante aux ordinateurs, est né le 28 décembre 1852 à Santa Cruz (Espagne). Il étudia à l'institut de Bilbao et à l'école d'ingénieurs de Madrid, avant de se lancer dans une carrière d'ingénieur et d'inventeur.

Ses talents d'inventeur ne se manifestèrent qu'à la fin de sa vie. Il conçut le téléphérique de la Niagara Transport Bridge (encore utilisé aujourd'hui aux chutes du Niagara), et un dirigeable semi-rigide qui fut fabriqué au cours de la Première Guerre mondiale. Mais, fondamentalement, Torres était un homme de son époque, et son intérêt se situait principalement dans les dispositifs électromécaniques. En 1906, il présenta au roi d'Espagne un modèle de bateau commandé par radio. En 1911, il inventa le premier joueur d'échecs automatique. La machine utilisait des électroaimants placés sous la table pour déplacer les pièces. Elle était programmée pour remporter une partie simple contre un adversaire humain.

Virgules flottantes

Comme beaucoup de ses contemporains, Torres était vraiment polyvalent. Ses recherches allèrent de la conception et de la construction de dispositifs mécaniques comme le dirigeable illustré ici, à la conception de machines à calculer électromécaniques. (Cl. Société aéronautique.)



L'intérêt de Torres pour les automates provenait de son expérience des chaînes de montage des usines du début du XX^e siècle. Toute sa vie, il tenta de séparer les problèmes qui nécessitaient l'intervention du cerveau humain de ceux qui pouvaient être résolus automatiquement.

En 1914, il publia un travail démontrant la possibilité de construire la machine analytique de Babbage en utilisant des techniques électromagnétiques. Dans ce rapport, il suggéra pour la première fois d'utiliser l'arithmétique en virgule flottante dans un ordinateur. En 1920, il construisit un calculateur électromagnétique, à partir d'une machine à écrire modifiée, pour

entrer les nombres et imprimer les résultats automatiquement. La machine à écrire était connectée au calculateur au moyen de fils téléphoniques. Torres pressentit la possibilité de relier des terminaux à un calculateur central.

L'Académie des sciences française reconnut la valeur de ses travaux et il devint plus tard président de l'Académie des sciences de son propre pays. Il mourut en 1936.

Arithmétique en virgule flottante

Une caisse enregistreuse affiche les totaux en francs et en centimes, et, dans une telle machine, deux positions seulement sont nécessaires après la virgule. Mais, dans un ordinateur, une précision plus grande est souvent nécessaire et le nombre de positions décimales peut varier ou « flotter » selon les besoins du problème. D'où l'« arithmétique en virgule flottante ».

Tout nombre peut être écrit de diverses façons. Par exemple, 0,875 2 m peut être exprimé comme 875,2 mm, ou $0,875 2 \times 1\,000$ mm, ou simplement $0,875 2 \times 10^3$. Cette dernière méthode permet une méthode de codage économique pour un ordinateur. Si un ordinateur ne dispose que de six positions pour représenter chaque nombre (et, pour simplifier les choses, le système décimal est utilisé à la place du système binaire), le nombre ci-dessus peut être stocké comme 875203; les deux derniers chiffres à droite sont nommés « index » et représentent la puissance de 10 (ici 3), et les quatre premiers chiffres forment la « mantisse ». Pour donner un autre exemple pour cet ordinateur : le nombre 418302 représente $0,418 3 \times 10^2$, ou 41,83.

La mantisse et l'index sont généralement « normalisés » en enlevant tous les zéros en tête de la mantisse. Par exemple, le nombre 41,83 pourrait être écrit 004104, mais serait normalisé à 418302 — incluant donc plus de chiffres significatifs dans la mantisse.

Cette forme index/mantisse de l'arithmétique en virgule flottante a l'avantage de pouvoir représenter une grande variété de nombres. Pour l'ordinateur suggéré plus haut, qui n'affecte que deux chiffres à l'index, un nombre aussi grand que $0,999 9 \times 10^{99}$ ou aussi petit que celui ayant 48 zéros après la virgule pourrait être manipulé.

Pendant, la précision de ce système demeure limitée aux chiffres affectés à la mantisse. Par conséquent, certains nombres ne peuvent qu'être représentés approximativement, et une grande attention et beaucoup d'ingéniosité doivent être employés dans les techniques de programmation arithmétiques pour éviter l'introduction d'erreurs. C'est pourquoi $(1/3) \times 3$ donne 0,999 999 9 sur certains ordinateurs alors que la réponse devrait être 1.

PARLONS L'ANGLAIS

COURS PROGRAMMÉ DE LANGUE ANGLAISE



EDITIONS
ATLAS

Protégés par une élégante reliure, vos numéros d'aBc Informatique seront plus faciles à consulter

Pour classer, répertorier, protéger vos fascicules d'ABC Informatique, les Éditions Atlas vous proposent des reliures élégantes, sobres, qui s'insèrent parfaitement dans votre bibliothèque. Chacune contient 12 fascicules,

les maintient, les préserve. Un système simple, résistant, vous permet de les assembler facilement. Elles sont en vente en permanence chez votre marchand de journaux. Demandez-les !



Chaque
reliure:

40 FF
295 FB
18 FS

Cod. N.M.P.P. :
6103

EDITIONS
ATLAS